BACHELOR'S THESIS

Degree in Mathematics

Title: Theoretical Study of Artificial Neural Networks

Author: Felipe Cano Córdoba

Advisor: Brian Subirana

Department: Mechanical Engineering

Academic year: 2017-2018

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

Universitat Politècnica de Catalunya
Facultat de Matemàtiques i Estadística

Degree in Mathematics

Bachelor's Degree Thesis

Theoretical Study of Artificial Neural Networks

Felipe Cano Córdoba

Supervised by Brian Subirana

June 2018

Abstract

The basic structure and definitions of artificial neural networks are exposed, as an introduction to Machine Learning algorithms. The theoretical description is emphasized and representation power of both shallow and deep networks is studied, proving the so called *Universality Theorem*. Then the properties and limitations of learning algorithms are studied. More specifically, the *No Free Lunch Theorem* is presented and proven, and then some recent approaches to the open problem of convergence of Stochastic Gradient Descent applied to neural networks are presented. Finally, a concept of forgetting in neural networks is introduced and some results on this model are given throughout the thesis.

Keywords

Artificial Neural Networks, Machine Learning, Deep Learning, Stochastic Gradient Descent, Universality Theorem, No Free Lunch Theorem, Forgetting Networks

Contents

Chapter 1

Introduction

1.1 Definition of Neural Networks

There are many different types of neural networks that have proved to be successful in solving different problems in the field of artificial intelligence.

We will define a quite general model for feedforward neural networks. Many specific cases, like multilayer perceptrons and convolutional networks can be regarded as particular cases of our model.

We will define neural networks in terms of its fundamental units, which we call neurons, as defined next.

Definition 1.1.1 (Neuron). Given a domain $\Omega \subseteq \mathbb{R}^n$, a ***neuron*** is a function $\eta : \Omega \to \mathbb{R}$ of the form

$$\eta(\boldsymbol{x}) = \sigma\big(\langle \boldsymbol{x}, \boldsymbol{w} \rangle + b\big) \tag{1.1.1}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ are the ***weights***, $b \in \mathbb{R}$ the ***bias*** and $\sigma : \mathbb{R} \to \mathbb{R}$ the ***activation function***.

Note on the output layer: We have omitted the possibility of the output layer having more than one output for the sake of simplicity. Networks with m output variables can be thought as having m different networks of one output each, and most of our results apply. This issue will be addressed for specific cases when necessary.

Definition 1.1.2 (Shallow network). Given a domain $\Omega \subseteq \mathbb{R}^n$ and an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, a ***shallow network*** of N ***units*** is a linear combination of N neurons, i.e. it is a function $\Sigma : \Omega \to \mathbb{R}$ of the form:

$$\Sigma(\boldsymbol{x}) = \sum_{k=1}^{N} a_k \sigma\big(\langle \boldsymbol{x}, \boldsymbol{w}_k \rangle + b_k\big) \tag{1.1.2}$$

where $a_k \in \mathbb{R}$.

Definition 1.1.3. We will denote the set of all shallow networks for a given σ, $\Omega \in \mathbb{R}^n$ and number of units N as

$$\mathcal{S}_{N,n}(\sigma, \Omega) \stackrel{\text{def}}{=} \left\{ \sum_{k=1}^{N} a_k \sigma\big(\langle \boldsymbol{x}, \boldsymbol{w}_k \rangle + b_k\big) \ : \ \boldsymbol{w}_k \in \mathbb{R}^n,\ a_k, b_k \in \mathbb{R} \right\} \tag{1.1.3}$$

and the set of all shallow networks as

$$\mathcal{S}_n(\sigma, \Omega) \stackrel{\text{def}}{=} \bigcup_{N=1}^{\infty} \mathcal{S}_{N,n}(\sigma, \Omega) \tag{1.1.4}$$

although we will usually drop the activation function and the domain and use $\mathcal{S}_{N,n}$ and $\mathcal{S}_n$ instead of $\mathcal{S}_{N,n}(\sigma, \Omega)$ and $\mathcal{S}_n(\sigma, \Omega)$.

Note that a shallow network of N units has $(n+2)N$ (trainable) parameters.

To talk about deep networks we need to define functions from graphs.

Definition 1.1.4 ($\mathcal{G}-$function). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected directed acyclic graph (CDAG), being $\mathcal{V}$ and $\mathcal{E}$ the sets of vertices and edges respectively, with n source nodes and one sink node. For any $v \in \mathcal{V}$, let d_v be the number of in-edges of v. Consider that each $v \in \mathcal{V}$ has an associated function $f_v : \mathbb{R}^{d_v} \to \mathbb{R}$ (we call this the ***constituent function*** of v). Let $\Omega \subseteq \mathbb{R}^n$ be a domain. A $\mathcal{G}-$***function*** is a function $G : \Omega \to \mathbb{R}$ that is computed by the following rule:

- Each source node is a real variable input (since there are n nodes, the domain is in $\mathbb{R}^n$).
- In any other node v, each of the in-edges represents a real variable input, the node computes the result of its constituent function f_v, the result is thrown as an input to the vertices in each out-edges of v.
- The result of the whole network is the output of the only sink node.

Note that two different sets of constituent functions for the same CDAG $\mathcal{G}$ can give rise to the same $\mathcal{G}$-function.

Deep networks' graphs can generally be divided into groups of nodes, corresponding to layers. Each layer receives input only from the previous layer. We give a formal definition of graphs like this:

Definition 1.1.5 (Layered graph). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a CDAG. Let $V_1 \subset \mathcal{V}$ be the set of source nodes, and for every integer $k > 1$, let $V_k \subset \mathcal{V}$ be the set of nodes that have an in-edge from a node in V_{k-1}. $\mathcal{G}$ is a ***layered graph*** if for every pair of integers $i \neq j$, $V_i \cap V_j = \emptyset$

Note that if for some k, $V_k = \emptyset$, then it follows from the definition that for all $n \geq k$ $V_n = \emptyset$.

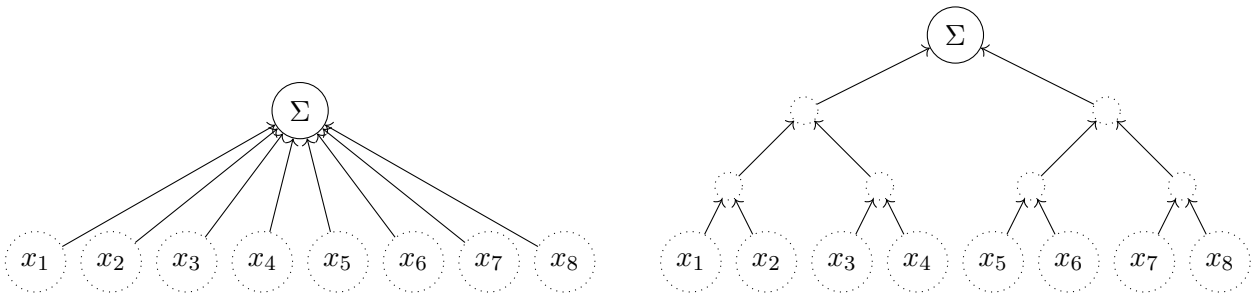


Figure 1.1: Shallow network *vs* binary-tree deep network.

If $\mathcal{G}$ is layered, $\exists d \in \mathbb{N}$ such that $|V_d| = 1$ and $V_{d+1} = \emptyset$. In this case, $\mathcal{V} = \bigsqcup_{i=1}^{d} V_i$.
Each of these sets of vertices is called a ***layer***. V_1 is called the ***input layer***, V_d is called the ***output layer*** and the rest are called ***hidden layers***.

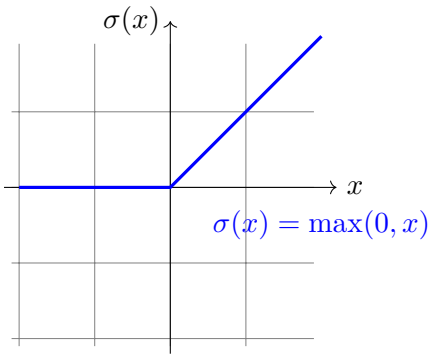
Definition 1.1.6 (Deep network). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a CDAG with n source nodes and one sink node. Let $\Omega \subseteq \mathbb{R}^n$ be a domain. A ***$\mathcal{G}$-deep network*** is a $\mathcal{G}-$function $\Delta : \Omega \to \mathbb{R}$ with constituent functions being all shallow networks.

One of the reasons for the interest in deep networks is that in most real world scenarios, functions have a $\mathcal{G}$-function structure [30, Appendix 2]. The reasoning comes from physics where it does not make sense that constituent functions are so pathological as the bijective functions between $\mathbb{R}$ and $\mathbb{R}^n$, and therefore our interest is focused in the internally continuous (or $\mathcal{C}^k$) case.

1.1.1 Note on Activation Function

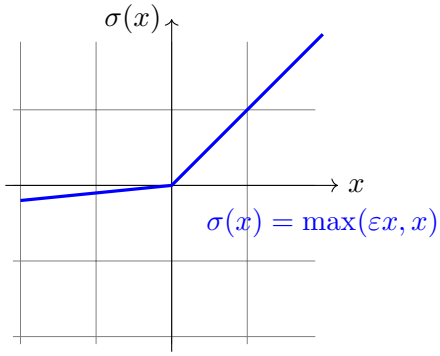
Most of the results we present depend on some characteristics of the activation functions σ, generally regarding its regularity. We want to shortly comment on the main activation functions we have found in the literature, as well as its regularity properties.

Rectified Linear Unit (ReLU)



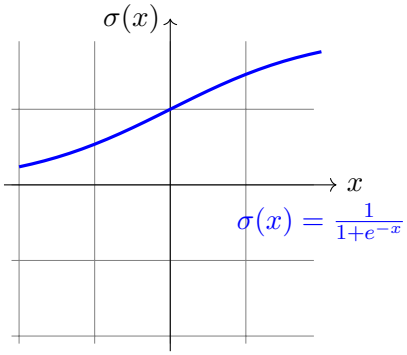
(a) The ReLU is probably the most widely used activation function. It is continuous, but not differentiable at $x = 0$.

Leaky ReLU



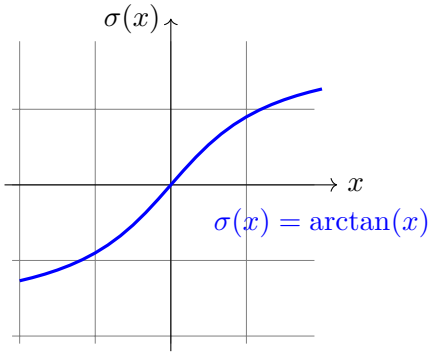
(b) This one shares regularity properties with the ReLU, but has the advantage of having non-vanishing derivative, useful in gradient based learning methods.

Sigmoid function



(a) Widely used. Infinitely differentiable and analytic.

Inverse Tangent



(b) Widely used. Infinitely differentiable, non analytic.

1.2 Neuroscience Insights and Forgetting Theory

We present a novel theory of how can neural networks mimic the animal brain forgetting behavior.

We do so by proposing a simple modification of the networks:

Definition 1.2.1 (Forgetting shallow network). Given a domain $\Omega \subseteq \mathbb{R}^n$, an activation function $\sigma : \mathbb{R} \to \mathbb{R}$, forgetting functions $\varphi_a, \varphi_w, \varphi_b : [0, \infty) \to \mathbb{R}$ a ***forgetting shallow network*** of N units is a function $\Sigma : \Omega \times [0, \infty) \to \mathbb{R}$ of the form:

$$\Sigma(\boldsymbol{x}; t) = \sum_{k=1}^{N} a_k \varphi_a(t) \sigma\big(\langle \boldsymbol{x}, \boldsymbol{w}_k \varphi_w(t) \rangle + b_k \varphi_b(t)\big) \tag{1.2.1}$$

where $a_k, b_k \in \mathbb{R}$, $\boldsymbol{w}_k \in \mathbb{R}^n$ are the network parameters.

The typical forgetting function is a decreasing exponential, but from a mathematical standpoint, any decreasing function $\varphi(t)$ with $\varphi(0) = 1$ and $\lim_{t\to\infty} \varphi(t) = 0$ can work as a forgetting function. Regular networks are recovered when all forgetting functions are considered to be constant ($\varphi(t) = 1$ for all t). We will refer to results being under the ***forgetting hypothesis*** when we want to emphasize that forgetting networks are considered.

In this definition we consider three individual forgetting functions, depending on the forgotten parameter involved (φ_a, φ_w and φ_b). Unless stated otherwise, along the thesis we will suppose that forgetting occurs outside of the activation function (and therefore $\varphi_w(t) = \varphi_b(t) = 1$ for all t).

Definition 1.2.2 (Forgetting deep network). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a CDAG with n source nodes and one sink node. Let $\Omega \subseteq \mathbb{R}^n$ be a domain. A ***$\mathcal{G}$-forgetting deep network*** is a $\mathcal{G}-$function $\Delta : \Omega \to \mathbb{R}$ with constituent functions being all forgetting shallow networks.

The motivation for these concepts comes from neuroscience, specifically from the works of Ebbinghaus [11].

Along the thesis we will comment how different results are generalized to this forgetting model.

Part I

Representation Power of Neural Networks

Chapter 2

Mathematical Background

2.1 Normed Spaces

The goal of all learning algorithms is to start from a given function f and some information about this function, and then find a function f^* that is "similar enough" to the original function.

The statement becomes precise when we define a norm in the space of functions. In that case we can study the value $\|f - f^*\|$ to evaluate to what extent f^* is "similar enough" to f and if possible find an optimal within our representation abilities.

Definition 2.1.1 (Normed space). A ***norm*** in a vector space $\mathbb{X}$ is a function $\|\cdot\| : \mathbb{X} \to \mathbb{R}$ with the following properties:

i $\|x\| \geq 0$ for all $x \in \mathbb{X}$ and $\|x\| = 0 \iff x = 0$

ii $\|\lambda x\| = |\lambda|\|x\|$ for all $\lambda \in \mathbb{R}$ and all $x \in \mathbb{X}$

iii $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in \mathbb{X}$ (triangular inequality)

The pair $(\mathbb{X}, \|\cdot\|)$ is called a ***normed space***.

Definition 2.1.2 (p-norm, $\mathcal{L}^p$ spaces). Let $\Omega \in \mathbb{R}^n$ be a domain, $p \in [1, \infty)$. Consider $f : \Omega \to \mathbb{R}$, then its p-***norm*** is (if it exists):

$$\|f\|_p \stackrel{\text{def}}{=} \left(\int_\Omega |f(\boldsymbol{x})|^p d\boldsymbol{x} \right)^{1/p} \tag{2.1.1}$$

This notion lets us define $\mathcal{L}^p$ spaces as:

$$\mathcal{L}^p(\Omega) \stackrel{\text{def}}{=} \{f : \Omega \to \mathbb{R} \ : \ \|f\|_p < \infty\} \tag{2.1.2}$$

For the case $p = \infty$, an analogous definition can be made, using the concept of essential supremum

$$\|f\|_\infty \stackrel{\text{def}}{=} \operatorname*{ess\,sup}_\Omega f = \inf\{a \in \mathbb{R} \ : \ f^{-1}(a, +\infty) \text{ is a set of measure zero in } A\} \tag{2.1.3}$$

The same definition can apply for $0 < p < 1$, but the resulting space is not a normed space. For $p \leq 0$, the problem is even worse because the norm is not defined for elementary functions like $f(x) = 0$ or $f(x) = x$.

Definition 2.1.3 ($\mathcal{C}^k$ spaces). Let $\Omega \subseteq \mathbb{R}^n$ and $k \in \mathbb{Z}_+$, we define the spaces

$$\mathcal{C}^k(\Omega) \stackrel{\text{def}}{=} \{f : \Omega \to \mathbb{R} \ : \ f \text{ has continuous partial derivatives up to order } k\} \tag{2.1.4}$$

$$\mathcal{C}^\infty(\Omega) \stackrel{\text{def}}{=} \bigcap_{k=1}^{\infty} \mathcal{C}^k(\Omega) \qquad ; \qquad \mathcal{C}(\Omega) \stackrel{\text{def}}{=} \{f : \Omega \to \mathbb{R} \ : \ f \text{ is continuous}\} \tag{2.1.5}$$

In all these spaces it is common the sup norm can be defined and it corresponds to the $\mathcal{L}^\infty$ norm for continuous functions.

Definition 2.1.4 (Lipschitz continuity). A function $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}^m$ is ***Lipschitz continuous*** if there exists a constant C such that for all $x, y \in \Omega$ it is satisfied that $\|f(x) - f(y)\| \leq C\|x - y\|$.

If Ω is compact, this condition is stronger than continuity, but weaker than continuously differentiable. If $\Omega = \mathbb{R}^n$, there exist $\mathcal{C}^\infty$ functions that are not Lipschitz continuous, for example $f(x) = x^2$.

2.1.1 Basic behavior of Forgetting Networks

Proposition 2.1.1 (Theorem of Non-Instantaneous Forgetting). *Let Δ be a forgetting deep network defined over the compact domain Ω with activation function $\sigma \in \mathcal{C}^1(\mathbb{R})$. Show that*

$$\lim_{t \to 0} \|\Delta(\,\cdot\,; t)\| = \Delta(\,\cdot\,; 0) \tag{2.1.6}$$

uniformly, when the norm used is the sup norm.

PROOF.
For a given $\boldsymbol{x} \in \Omega$, we define the function $D_{\boldsymbol{x}}(t) \stackrel{\text{def}}{=} \Delta(\boldsymbol{x}; t)$.

In that case, $D_{\boldsymbol{x}}(t)$ is continuous with respect to t because it is the composition of continuous functions (because the only dependence on t is on the functions $\varphi(t)$, which are clearly continuous by hypothesis). If $\varphi(t)$ could jump discontinuously to 0, forgetting would too.

Given ε, continuity of $D_{\boldsymbol{x}}(t)$ implies $\forall \boldsymbol{x} \in \Omega$, $\exists \delta t$, that depends on $\boldsymbol{x}$ and ε, such that

$$\forall t \in (-\delta t, \delta t) \qquad |D_{\boldsymbol{x}}(0) - D_{\boldsymbol{x}}(\delta t)| < \varepsilon \tag{2.1.7}$$

if we look now at $D_{\boldsymbol{x}}(t)$ as a function of $\boldsymbol{x}$ (with fixed t), since σ is continuously differentialbe, $D_x(t)$ is Lipshitz continuous with respect to $\boldsymbol{x}$. This means that for all $t \in \mathbb{R}$, there exists $C_t > 0$ such that $|D_{\boldsymbol{x}}(t) - D_{\boldsymbol{y}}(t)| \leq C_t \|\boldsymbol{x} - \boldsymbol{y}\|$.

With these tools, we will prove that given $\varepsilon > 0$, for all $\boldsymbol{x} \in \Omega$, there exists $r_{\boldsymbol{x}} > 0$ and δt, such that for all $\boldsymbol{y} \in \Omega$ and $t \in \mathbb{R}$

$$(\|x - y\| < r_{\boldsymbol{x}} \text{ and } |t| < \delta t) \implies |D_{\boldsymbol{y}}(0) - D_{\boldsymbol{y}}(t)| < \varepsilon \tag{2.1.8}$$

Continuity of $D_{\boldsymbol{x}}(t)$ with respect to t states that, considering $\varepsilon/3$, , there exists δt fulfilling equation (2.1.7). Let C_t be the Lipshitz constant of $D_{\boldsymbol{x}}(t)$ as stated before, we define $C = \max_{t \in [-\delta t, +\delta t]} \{C_t\}$. It is satisfied:

$$\forall t \in [-\delta t, +\delta t] \qquad |D_{\boldsymbol{x}}(t) - D_{\boldsymbol{y}}(t)| \leq C \|\boldsymbol{x} - \boldsymbol{y}\| \tag{2.1.9}$$

Now if we define $r_{\boldsymbol{x}} = \frac{\varepsilon}{3C}$, we have, for all $\boldsymbol{y}, t$ such that $\|\boldsymbol{x} - \boldsymbol{y}\| < r_{\boldsymbol{x}}$ and $|t| < \delta t$:

$$|D_{\boldsymbol{y}}(0) - D_{\boldsymbol{y}}(t)| = |D_{\boldsymbol{y}}(0) - D_{\boldsymbol{x}}(0) + D_{\boldsymbol{x}}(0) - D_{\boldsymbol{x}}(t) + D_{\boldsymbol{x}}(t) - D_{\boldsymbol{y}}(t)| \tag{2.1.10}$$

$$\leq |D_{\boldsymbol{y}}(0) - D_{\boldsymbol{x}}(0)| + |D_{\boldsymbol{x}}(0) - D_{\boldsymbol{x}}(t)| + |D_{\boldsymbol{x}}(t) - D_{\boldsymbol{y}}(t)| \quad \text{Triangular inequality} \tag{2.1.11}$$

$$\leq C\|\boldsymbol{x} - \boldsymbol{y}\| + \varepsilon/3 + C\|\boldsymbol{x} - \boldsymbol{y}\| \quad \text{Equations (2.1.9) and (2.1.7)} \tag{2.1.12}$$

$$\leq C r_{\boldsymbol{x}} + \varepsilon/3 + C r_{\boldsymbol{x}} = C\frac{\varepsilon}{3C} + \frac{\varepsilon}{3} + C\frac{\varepsilon}{3C} = \varepsilon \quad \text{Definition of } r_{\boldsymbol{x}} \tag{2.1.13}$$

This proves equation (2.1.8). Now we will use the fact that Ω is compact. Consider the open cover of Ω:

$$\Omega = \bigcup_{\boldsymbol{x} \in \Omega} B(\boldsymbol{x}, r_{\boldsymbol{x}}) \tag{2.1.14}$$

where $B(\boldsymbol{x}, r_{\boldsymbol{x}})$ is the open ball centered in $\boldsymbol{x}$ and radius $r_{\boldsymbol{x}}$ as defined before. Since Ω is compact, there exists $N \in \mathbb{N}$ and a finite number of elements $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N$ such that

$$\Omega = \bigcup_{i=1}^{N} B(\boldsymbol{x}_i, r_{\boldsymbol{x}_i}) \tag{2.1.15}$$

For each $\boldsymbol{x}_i$ consider $(\delta t)_{\boldsymbol{x}_i}$ that fulfills equation (2.1.8). We define [1] $\delta t = \min_{i=1:N} \{(\delta t)_{\boldsymbol{x}_i}\}$.

The proof is finished because given $\varepsilon > 0$, δt as we have just defined fulfills

$$|\Delta_{\boldsymbol{x}}(0) - \Delta_{\boldsymbol{x}}(t)| < \varepsilon \qquad \forall \boldsymbol{x} \in \Omega \text{ if } |t| < \delta t \tag{2.1.16}$$

Since we are using the *sup* norm, this is directly what we want to prove. ■

[1] In this step the importance of Ω being compact becomes clear. If it wasn't, the minimum δt may be zero.

Proposition 2.1.2 (Theorem of Universal Forgetting). *Let Δ be a forgetting deep network defined over the compact domain Ω with activation function $\sigma \in \mathcal{C}(\mathbb{R})$. Show that*

$$\lim_{t\to\infty} \|\Delta(\,\cdot\,;t)\| = 0 \tag{2.1.17}$$

uniformly, when the norm used is the sup norm.

PROOF.
This result is a direct consequence of continuity of φ and σ. We follow by induction on N the number of nodes of the graph. The base case is $n = 2$, in which case the only dependence on t if $\varphi(t)$ multiplying the network, so clearly the limit is zero.

Let $\mathcal{G}$ be the graph associated to Δ and let f be the constituent function of the only sink node.

$$f(x_1,\ldots,x_{d_s};t) = \sum_{i=1}^{N_s} a_k\varphi(t)\sigma\big(\langle \boldsymbol{x}, \boldsymbol{w}_k\rangle + b_k\big) \tag{2.1.18}$$

For each in-edge of the sink node, consider v_k the vertex the edge is coming from. Since the degree of the sink node is d_s, there are d_s different vertices. For each of these vertices, consider $\mathcal{G}_k$ the maximal subgraph of $\mathcal{G}$ such that $\mathcal{G}_k$ is a CDAG with v_k as its only sink node. If we consider the same constituent functions as in $\mathcal{G}$ this constructions gives rise to deep networks $\Delta_1,\ldots,\Delta_{d_s}$ each one of them with less nodes than the original network. We will apply the induction hypothesis there. For any $\boldsymbol{x} \in \mathbb{R}^n$, let us denote the vector $\boldsymbol{X}(\boldsymbol{x};t) = (\Delta_1(\boldsymbol{x}_1;t),\ldots,\Delta_{d_s}(\boldsymbol{x}_{d_s};t))$. With this notation:

$$\Delta(\boldsymbol{x};t) = \sum_{k=1}^{N_s} a_k\varphi(t)\sigma\Big(\big\langle \boldsymbol{X}, \boldsymbol{w}_k\big\rangle + b_k\Big) \tag{2.1.19}$$

The induction hypothesis implies that $\lim_{t\to 0} \boldsymbol{X}(\boldsymbol{x};t) = \boldsymbol{0}$. Taking limits when $t \to 0$ in $\Delta(\boldsymbol{x};t)$:

$$\begin{aligned}
\lim_{t\to 0}\Delta(\boldsymbol{x};t) &= \sum_{k=1}^{N_s} a_k \lim_{t\to 0}\varphi(t) \lim_{t\to 0}\sigma\Big(\big\langle \boldsymbol{X}, \boldsymbol{w}_k\big\rangle + b_k\Big) && \text{Linearity of limits, if both exist}\\
&= \sum_{k=1}^{N_s} a_k \lim_{t\to 0}\varphi(t)\sigma\Big(\big\langle \lim_{t\to 0}\boldsymbol{X}, \boldsymbol{w}_k\big\rangle + b_k\Big) && \text{Continuity of } \sigma\\
&= \sum_{k=1}^{N_s} a_k \lim_{t\to 0}\varphi(t)\sigma(b_k) && \text{Induction hypothesis}\\
&= \sum_{k=1}^{N_s} a_k \cdot 0 \cdot \sigma(b_k) = 0 && \Big(\lim_{t\to 0}\varphi(t) = 0\Big)
\end{aligned}$$

■

2.2 Sobolev Spaces

One technical detail to have in mind is that $\mathcal{L}^p$ as we have defined it is not rigorously a normed space, because there are non zero functions that have zero norm. To solve this, whenever two functions f, g satisfy $\|f-g\|_p = 0$, they will be considered the same function in $\mathcal{L}^p$.

This happens when $(f-g)^p = 0$ almost everywhere, that is equivalent to $f - g = 0$ almost everywhere. As a consequence if $\|f-g\|_p = 0$ for some p, then for all $q \in [1, \infty]$ $\|f-g\|_q = 0$. That is, whenever two functions f, g are considered the same in some $\mathcal{L}^p$ space, they are also considered the same in any other $\mathcal{L}^q$.

Since in $\mathcal{L}^p$ space the notion of the value of a function in a point has no meaning (a point is of measure zero), there is a priori no notion of derivative. This problem is solved by defining a suitable concept of weak derivative, that extends its classical version.

Consider two functions $F, \varphi \in \mathcal{C}^1(\mathbb{R})$, for some domain $I = [a, b] \subseteq \mathbb{R}$, it is well known that (integration by parts formula):

$$\int_a^b \frac{\partial F}{\partial x}\varphi = [F\varphi]_{x=a}^{x=b} - \int_a^b F\frac{\partial \varphi}{\partial x} \tag{2.2.1}$$

If we consider $\varphi \in \mathcal{C}_0^\infty(I)$, it is satisfied that $[F\varphi]_{x=a}^{x=b} = 0$, then the formula above becomes:

$$\int_a^b \frac{\partial F}{\partial x}\varphi = -\int_a^b F\frac{\partial \varphi}{\partial x} \tag{2.2.2}$$

If we consider all possible $\varphi \in \mathcal{C}_0^\infty(I)$, given $F \in \mathcal{L}^p(I)$, it can be shown that the formula:

$$\int_a^b \frac{\partial F}{\partial x}\varphi = -\int_a^b F\frac{\partial \varphi}{\partial x} \qquad \forall \varphi \in \mathcal{C}_0^\infty(I) \tag{2.2.3}$$

defines $\frac{\partial F}{\partial x}$ in the sense that it may or may not exist, but if it exists, $\frac{\partial F}{\partial x}$ is unique.

This can be generalized to n variables and derivatives of order k as follows.

Definition 2.2.1 (Weak derivative). Let $\Omega \subseteq \mathbb{R}^n$ be a domain, $F \in \mathcal{L}^p(\Omega)$. then the $\boldsymbol{k}$-th derivative of F can be defined (if it exists) as the only function satisfying:

$$\int_\Omega D^{\boldsymbol{k}} F \cdot \varphi = (-1)^{|\boldsymbol{k}|} \int_\Omega F \cdot D^{\boldsymbol{k}}\varphi \qquad \forall \varphi \in \mathcal{C}_0^\infty(\Omega) \tag{2.2.4}$$

where $\boldsymbol{k}$ is the multi-integer $\boldsymbol{k} = (k_1, \ldots, k_n)$, $|\boldsymbol{k}| = \sum_{i=1}^n k_i$ and $D^{\boldsymbol{k}} f = \dfrac{\partial^{|\boldsymbol{k}|} f}{\partial^{k_1} x_1 \cdots \partial^{k_n} x_n}$.

For a discussion on the existence and properties of weak derivatives, see [3, Ch. 3].

Example 1. Consider $f(x) = \begin{cases} 0 & \text{if } x \in \mathbb{Q} \\ \sin x & \text{if } x \notin \mathbb{Q} \end{cases}$ Since $\mathbb{Q}$ measure zero, this function is equal to $\tilde{f}(x) = \sin x$ in any $\mathcal{L}^p$, and as a differentiable function its derivative is $\cos x$.

Example 2. Another typical example is $f(x) = |x|$. This function has no classical derivative in $x = 0$. In this case it can be shown that the weak derivative is the sign function:

$$\sigma(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases}$$

This concept of weak derivative gives rise to the definition of Sobolev spaces, which are normed spaces with a certain number of (weak) derivatives.

Definition 2.2.2 (Sobolev norm). Let $f \in \mathcal{L}^p(\Omega)$. The ***Sobolev norm*** of a function $f \in \mathcal{L}^p(\Omega)$ defined (if all weak derivatives exist) as:

$$\|f\|_{p,m} \stackrel{\text{def}}{=} \sum_{0 \leq |\boldsymbol{k}| \leq m} \|D^{\boldsymbol{k}} f\|_p \tag{2.2.5}$$

Note that this sum has $\binom{n+m}{n}$ terms.

Definition 2.2.3 (Sobolev spaces). Given the space of functions $\mathcal{L}^p(\Omega)$, $\Omega \subseteq \mathbb{R}^n$. A ***Sobolev space*** in $\mathbb{R}^n$ with Sobolev norm $\|\cdot\|_{m,p}$ is the set of all functions $f : \mathbb{R}^n \to \mathbb{R}$ that can be weakly derived according to the multi integer $\boldsymbol{k}$ when $|\boldsymbol{k}| \leq m$ and that have a Soolev norm smaller than one. Formally:

$$W^n_{p,m}(\Omega) \stackrel{\text{def}}{=} \{f \in \mathcal{L}^p(\Omega) \ : \ \|f\|_{p,m} \leq 1\} \tag{2.2.6}$$

This ensures that for a function in the Sobolev space with m derivatives, this function must have derivatives up to order m in $\mathcal{L}^p$ and consequently derivatives do not get "too large".

2.3 Convolution and Mollifiers

In this section we introduce two key concepts in mathematical analysis, that we will need for the proofs of many theorems in the following chapters. To do so, we first need to introduce some notation.

Definition 2.3.1 (Support, $\mathcal{C}_0^\infty(\Omega)$). Let f be a function defined in some domain $\Omega \subseteq \mathbb{R}^n$, the ***support*** of f is defined as:

$$\operatorname{supp}(f) \stackrel{\text{def}}{=} \Omega \cap \operatorname{closure}\{\boldsymbol{x} \in \Omega \ : \ f(\boldsymbol{x}) \neq 0\} \tag{2.3.1}$$

We say that a function f has compact support if its support is a compact subset of Ω. The set of k times differentiable functions will be denoted as:

$$\mathcal{C}_0^k(\Omega) \stackrel{\text{def}}{=} \{f \in \mathcal{C}^k(\Omega) \ : \ \operatorname{supp}(f) \text{ is a compact set}\} \tag{2.3.2}$$

It will be of special interest the case of $k = \infty$.

Definition 2.3.2 (Ball). Let $\boldsymbol{x} \in \mathbb{R}^n$ and $r > 0$, we will denote $B(\boldsymbol{x}, r)$ the ***ball*** with center $\boldsymbol{x}$ and radius r. Formally

$$B(\boldsymbol{x}, r) \stackrel{\text{def}}{=} \{\boldsymbol{y} \in \mathbb{R}^n \ : \ |x - y| < r\} \tag{2.3.3}$$

Example 3. The typical example of a function with compact support is:

$$\eta(\boldsymbol{x}) = \begin{cases} c e^{\left(\frac{-1}{1-|\boldsymbol{x}|^2}\right)} & |\boldsymbol{x}| \leq 1 \quad (c \in \mathbb{R}) \\ 0 & |\boldsymbol{x}| > 1 \end{cases} \tag{2.3.4}$$

We can choose $c = \left(\int_{B(\mathbf{0},1)} e^{\left(\frac{-1}{1-|\boldsymbol{x}|^2}\right)} d\boldsymbol{x} \right)^{-1}$ so that the property $\int_{\mathbb{R}^n} \eta = 1$ is satisfied.

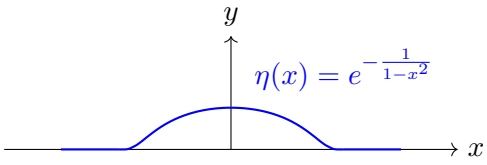


Figure 2.1: Graph of $\eta(x)$ for the unidimensional case

We can construct other examples from eq. (2.3.4). Given $\varepsilon < 0$,

$$\eta_\varepsilon(\boldsymbol{x}) \stackrel{\text{def}}{=} \frac{1}{\varepsilon^n} \eta\left(\frac{\boldsymbol{x}}{\varepsilon}\right) \tag{2.3.5}$$

has support $B(\mathbf{0}, \varepsilon)$ and also satisfies $\int_{\mathbb{R}^n} \eta_\varepsilon = 1$.

Definition 2.3.3 (Convolution). Let f, g be measurable functions in $\mathbb{R}^n$. The convolution of f and g is defined as

$$(f * g)(\boldsymbol{x}) \stackrel{\text{def}}{=} \int_{\mathbb{R}^n} f(\boldsymbol{y}) g(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y} \tag{2.3.6}$$

This integral may not exist depending on the shape of f and g. An interesting case, that we will use, is when $f \in \mathcal{C}(\mathbb{R})$ and $g \in \mathcal{C}_0^\infty(\mathbb{R})$. In that case $(f * g)(\boldsymbol{x})$ exists for all $\boldsymbol{x}$ and is infinitely differentiable by the following result

Lemma 2.3.1. *Given two measurable functions f, g*

i Convolution is commutative: $f * g = g * f$.

ii $f \in \mathcal{C}^j(\mathbb{R})$ and $g \in \mathcal{C}^k(\mathbb{R})$, then $f * g \in \mathcal{C}^{j+k}(\mathbb{R})$.

PROOF.
(i) This follows directly from a change of variable in the integral $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y}$:

$$(f * g)(\boldsymbol{x}) = \int_{\mathbb{R}^n} f(\boldsymbol{y}) g(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y} \tag{2.3.7}$$

$$= \int_{\mathbb{R}^n} f(\boldsymbol{x} - \boldsymbol{z}) g(\boldsymbol{z}) d(\boldsymbol{x} - \boldsymbol{z}) = (g * f)(\boldsymbol{x}) \tag{2.3.8}$$

(ii) This follows from the derivative property: $\partial_{x_i}(f * g) = (\partial_{x_i} f) * g$:

$$\partial_{x_i}(f * g) = \partial_{x_i} \left(\int_{\mathbb{R}^n} f(\boldsymbol{y}) g(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y} \right) \tag{2.3.9}$$

$$\int_{\mathbb{R}^n} f(\boldsymbol{y}) \partial_{x_i}(g(\boldsymbol{x} - \boldsymbol{y})) d\boldsymbol{y} \qquad f \text{ is constant with respect to } \boldsymbol{x} \tag{2.3.10}$$

$$\int_{\mathbb{R}^n} f(\boldsymbol{y}) (\partial_{x_i} g)(\boldsymbol{x} - \boldsymbol{y}) d\boldsymbol{y} = f * \partial_{x_i} g \qquad \partial_{x_i}(\boldsymbol{x} - \boldsymbol{y}) = Id \tag{2.3.11}$$

■

In particular, if either f or g is infinitely differentiable, $f * g$ becomes also infinitely differentiable, regardless of the smoothness of the other.

The functions η_ε are called ***mollifiers*** because they have the following property (see fig. 2.2 for a numerical example)

Lemma 2.3.2. *Let $f \in \mathcal{C}(\Omega)$ and $f_\varepsilon \stackrel{def}{=} f * \eta_\varepsilon$. Then:*

1. $\text{supp}(f_\varepsilon) \subseteq \{\boldsymbol{x} \in \mathbb{R}^n \ : \ dist(\boldsymbol{x}, \text{supp}(f)) < \varepsilon\}$
2. $f_\varepsilon \in \mathcal{C}^\infty(\Omega)$
3. $f_\varepsilon \xrightarrow[\varepsilon \to 0]{} f$ *uniformly in each compact $K \subseteq \mathbb{R}^n$.*

PROOF.
The proof is made following [32, Lemma 7.1].

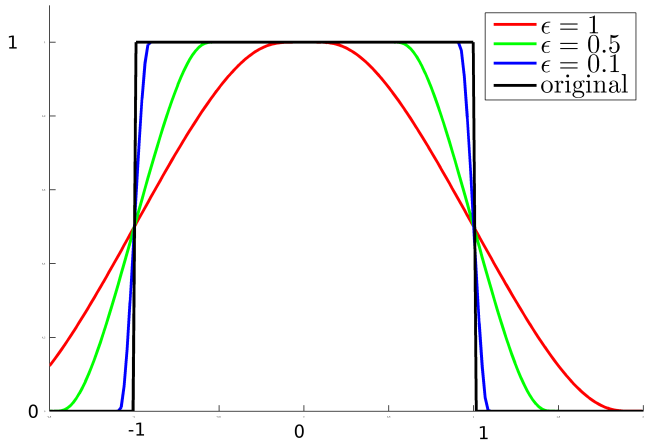


Figure 2.2: Graph of $H_\varepsilon(x)$ for different values of ε, being $H(x)$ the rectangle function: $H(x) = \begin{cases} 1 & \text{if } |x| \leq 1 \\ 0 & \text{if } |x| > 1 \end{cases}$

(i) Let $S = \operatorname{supp} f$. It suffices to prove that if $\operatorname{dist}(x, S) \geq \varepsilon$, then $f_\varepsilon(\boldsymbol{x}) = 0$.

$f_\varepsilon(\boldsymbol{x}) = \int_{\mathbb{R}^n} \eta_\varepsilon(\boldsymbol{z}) f(\boldsymbol{x} - \boldsymbol{z}) d\boldsymbol{z} = \int_{B(\mathbf{0},\varepsilon)} \eta_\varepsilon(\boldsymbol{z}) f(\boldsymbol{x} - \boldsymbol{z}) d\boldsymbol{z}$ because $\operatorname{supp}(\eta_\varepsilon) = B(\mathbf{0}, \varepsilon)$.

If $|\boldsymbol{z}| < \varepsilon$ and $\operatorname{dist}(\boldsymbol{x}, S) \geq \varepsilon$, triangular inequality states that $\operatorname{dist}(\boldsymbol{x} - \boldsymbol{z}, S) \geq \operatorname{dist}(\boldsymbol{x}, S) - \operatorname{dist}(\boldsymbol{z}, S) > 0$, then $\boldsymbol{x} - \boldsymbol{z} \notin \operatorname{supp} f$, so $f(\boldsymbol{x} - \boldsymbol{z}) = 0$. From the definition of f_ε as an integral with $f(\boldsymbol{x} - \boldsymbol{z})$ as a factor, directly follows $f_\varepsilon(\boldsymbol{x}) = 0$.

(ii) This is a direct consequence of lemma 2.3.1 because $\eta_\varepsilon \in \mathcal{C}_0^\infty$.

(iii) Since $\int_{\mathbb{R}^n} \eta_\varepsilon = 1$, we can write

$$f_\varepsilon(\boldsymbol{x}) - f(\boldsymbol{x}) = \int_{\mathbb{R}^n} \eta_\epsilon(\boldsymbol{z}) \left[f(\boldsymbol{x} - \boldsymbol{z}) - f(\boldsymbol{x}) \right] d\boldsymbol{z} \tag{2.3.12}$$

In this form we can see that for any $\boldsymbol{x} \in \Omega$, we have that

$$|f_\varepsilon(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \sup_{B(\mathbf{0},\varepsilon)} |f(\boldsymbol{x} - \boldsymbol{z}) - f(\boldsymbol{x})| \tag{2.3.13}$$

If we consider $K \subseteq \Omega$ compact, then f is uniformly continuous in K, therefore $\sup_{B(\mathbf{0},\varepsilon)} |f(\boldsymbol{x} - \boldsymbol{z}) - f(\boldsymbol{x})| \xrightarrow[\varepsilon \to 0]{} 0$ uniformly for $\boldsymbol{x} \in K$. This uniform convergence together with eq. (2.3.13) gives the result that $f_\varepsilon - f \xrightarrow[\varepsilon \to 0]{} 0$ uniformly in K. ■

2.4 Results of Approximation Theory

In this section we will present some concepts and results from approximation theory that will be used throughout the thesis, specially in chapter 4.

The first result is a classical one in real analysis, first proved in [10].

Lemma 2.4.1. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be an infinitely differentiable function. Then the following conditions are equivalent:*

i σ is not a polynomial.

ii There exists $x \in \mathbb{R}$ such that for all $n \in \mathbb{N}$ $\sigma^{(n}(x) \neq 0$.

PROOF.

$\boxed{i \implies ii}$

Let m be the degree of σ, then its $(m+1)$-th derivative vanishes, so $\forall x \in \mathbb{R}$, $\sigma^{(m+1}(x) = 0$.

$\boxed{ii \implies i}$

Let us define $E_n \stackrel{\text{def}}{=} \{x \in \mathbb{R} \ : \ f^{(n}(x) = 0\}$. This sets are closed due to the continuity of f and $\cup_{n\in\mathbb{N}} E_n = \mathbb{R}$ by hypothesis. We recall Baire's category theorem ([7]) which in our case states that for every numerable collection of dense open sets $\{U_n\}_{n\in\mathbb{N}} \subseteq \mathbb{R}$, their intersection is dense. In our case, let U_n be the complementary set of E_n: $U_n \stackrel{\text{def}}{=} E_n^c$. If all U_n were dense, by Baire's theorem, its intersection would be dense. But taking into account De Morgan laws

$$\bigcap_{n\in\mathbb{N}} U_n = \bigcap_{n\in\mathbb{N}} E_n^c = \left(\bigcup_{n\in\mathbb{N}} E_n\right)^c = \mathbb{R}^c = \emptyset \quad \text{(which is not dense in } \mathbb{R}) \tag{2.4.1}$$

As a consequence, there is one set U_n not dense, equivalently, one set E_n with non-empty interior, therefore it contains an interval $I \subseteq E_n$. So we have an interval I where $f^{(n}(x) = 0 \quad \forall x \in I$, meaning that f is a polynomial of degree at most n in I.

Now let Λ be a set of indexes, and $\{I_\lambda\}_{\lambda\in\Lambda}$ be the set of all maximal open intervals I_λ such that f is a polynomial in I_λ. We have already seen that there exists at least one such interval. We also observe that these intervals are mutually disjoint because they are maximal (if two I_λ and I_μ satisfy $I_\lambda \cap I_\mu \neq \emptyset$, then $I_\lambda \cup I_\mu$ is a larger interval).

Now we define

$$H \stackrel{\text{def}}{=} \mathbb{R} \setminus \bigcup_{\lambda\in\Lambda} I_\lambda \tag{2.4.2}$$

H has empty interior. If it did not, it would contain an interval $J \subseteq H$ and applying the same argument with Baire's theorem as previously to J instead of $\mathbb{R}$ we would find an interval $J' \subseteq J$ in which f is a polynomial, which generates a contradiction.

Now we prove that H has no isolated points. Suppose $x \in H$ is an isolated point. For this to happen, there would be two intervals $I_1, I_2 \in \{I_\lambda\}_{\lambda\in\Lambda}$ such that x is the right endpoint of I_1 and the left endpoint of I_2. There would then be also an integer n such that the n-th derivative vanishes in $I_1 \cup I_2$, and by continuity of $f^{(n}$ it would also vanish in x, and thus $I_1 \cup \{x\} \cup I_2$ is a larger interval in which f is a polynomial.

H is a closed subspace of $\mathbb{R}$, so if it is not empty, using Baire's theorem again [2], there exists and interval J such that $J \cap H \neq \emptyset$ and for some n

$$f^{(n}(x) = 0 \qquad \forall x \in J \cap H \tag{2.4.3}$$

Since H has non isolated points, any $x \in J \cap H$ is an accumulation point of $J \cap H$, so using only points in $J \cap H$ we can calculate

$$\lim_{h\to 0} \frac{f^{(n}(x+h) - f^{(n}(x)}{h} \tag{2.4.4}$$

This limit exists because f is infinitely differentiable and by construction it vanishes in $J \cap H$

$$f^{(n+1}(x) = 0 \qquad \forall x \in J \cap H \tag{2.4.5}$$

and repeating this argument,

$$f^{(m}(x) = 0 \qquad \forall x \in J \cap H \quad \forall m \geq n \tag{2.4.6}$$

Now we claim that there exists an interval $I \in \{I_\lambda\}_{\lambda\in\Lambda}$ contained in J. If there were no such interval, it would mean that the set $J \cap H$ contains an interval. But then H would contain an interval in which f is a polynomial, that should have been included in $\{I_\lambda\}_{\lambda\in\Lambda}$.

Let $I \subseteq J$ be such interval. Since f is a polynomial in I, there exists m such that $f^{(m}$ vanishes in I. Suppose $m > n$. Since the endpoints of I are in $J \cap H$ (therefore $f^{(m}$ vanishes at the mentioned endpoints) and $f^{(m} = 0$ in I, it is deduced that $f^{(m-1}$ vanishes in I. Applying induction we deduce that $f^{(n}$ vanishes in I.

Choose a point $x \in J \cap H$. We have seen that $J \cap H$ cannot contain an interval, so there must exist two intervals $I_1, I_2 \in \{I_\lambda\}_{\lambda\in\Lambda}$ such that x is the left endpoint of I_2 and the right endpoint of I_1. Since $f^{(n}$ is continuous and is zero at $I_1 \cup I_2$, it must be zero also in x, which is a contradiction because then f is a polynomial in $I_1 \cup \{x\} I_2$ which is a larger interval. This contradiction comes from assuming $H \neq \emptyset$, and f is then a polynomial in the only maximal interval $\mathbb{R}$. ■

Most of the following results will be based in the concept of approximation error that we introduce next:

[2] $E_n \cap H$ are closed subsets of H and $\bigcup_{n\in\mathbb{N}}(E_n \cap H) = H$, then there must be some of the E_n having non-empty interior in H.

Definition 2.4.1 (Approximation error). Let $(\mathbb{X}, \|\cdot\|)$ be a normed space, $W \subseteq \mathbb{X}$ and $f \in \mathbb{X}$, and W a subset of $\mathbb{X}$. The ***best approximation error*** of f in W is

$$E(f; W; \|\cdot\|) \stackrel{\text{def}}{=} \inf_{g \in W} \|f - g\| \tag{2.4.7}$$

Let $V \subseteq \mathbb{X}$. The ***worst case approximation error*** of V in W is

$$E(V; W; \|\cdot\|) \stackrel{\text{def}}{=} \sup_{f \in V} \inf_{g \in W} \|f - g\| \tag{2.4.8}$$

We will usually drop the norm when it is understood from the context which norm is considered. In that case we will write $E(f; W)$ instead of $E(f; W; \|\cdot\|)$.

First, a classical result from approximation theory:

Lemma 2.4.2. *Given P_k^n the space of polynomials of degree at most k in n variables and W_m^n the Sobolev space as defined in definition 2.2.3 with the sup norm, there exists a constant C such that the following inequality holds:*

$$E(\mathrm{W}_m^n; P_k^n) \leq Ck^{-m} \tag{2.4.9}$$

Since the proof is long and is not specially relevant, the reader is referred to section 2.4.1.

To prove that the complexity given is the best possible, we need to define the concepts of ***Bernstein N-width*** and ***continuous non-linear N-width*** (see [1] and [28, Sec. 6])

Definition 2.4.2 (Bernstein N-width). Given $\mathbb{X}$ a normed linear space and $K \subseteq \mathbb{X}$ a compact subset of it, the ***Bernstein N-width*** is:

$$b_N(K; X) \stackrel{\text{def}}{=} \sup_{X_{N+1}} \sup\{\lambda \,:\, \lambda S(X_{d+1}) \subseteq K\} \tag{2.4.10}$$

Where X_{N+1} is any $(N+1)$-dimensional subspace of $\mathbb{X}$ and $S(X_{N+1})$ is the unit ball of X_{N+1}.

Definition 2.4.3 (Continuous non-linear N-width). Given $\mathbb{X}$ a normed linear space and $K \subseteq \mathbb{X}$ a compact subset of it. Let $P_N : K \to \mathbb{R}^N$ be a continuous function and let $M_N : \mathbb{R}^N \to \mathbb{X}$ be any function. For each such P_N and M_N set

$$E(K; P_N, M_N; \mathbb{X}) \stackrel{\text{def}}{=} \sup_{f \in K} \|f - M_N(P_N(f))\| \tag{2.4.11}$$

and now define the ***continuous non-linear N-width*** as

$$h_N(K; \mathbb{X}) \stackrel{\text{def}}{=} \inf_{P_N, M_N} E(K; P_N, M_N; \mathbb{X}) \tag{2.4.12}$$

The idea behind this definition is the following:

- A learning algorithm can be regarded as a function $\Lambda : K \to \mathbb{X}$, because given a target function $f \in K$, returns its approximating neural network, which is a function in $\mathbb{X}$.
- This function Λ can be factorized into two functions $\Lambda : K \xrightarrow{P_N} \mathbb{R}^N \xrightarrow{M_N} \mathbb{X}$. P_N maps every function to a set of parameters, and given a set of parameters, M_N returns its corresponding neural network as a function in $\mathbb{X}$.
- Given an approximating algorithm (i.e. given P_N and M_N), $E(K; P_N, M_N; \mathbb{X})$ is the worst case error of the considered algorithm.
- Given K and $\mathbb{X}$, $h_N(K; \mathbb{X})$ represents the minimum $E(K; P_N, M_N; \mathbb{X})$ over all possible algorithms (P_N, M_N).

So our $E(K, \mathcal{S}_{n,N})$ is equal to $h_{(n+2)N}(K, \mathbb{X})$ when we restrict learning algorithms (P_N, M_N) to be P_N continuous and M_N the exact one described in the definition of shallow networks.

Lemma 2.4.3. *For any normed space $\mathbb{X}$ and $K \subseteq \mathbb{X}$ compact*

$$h_N(K; \mathbb{X}) \geq b_N(K; \mathbb{X}) \tag{2.4.13}$$

PROOF.
Let $P_N : K \to \mathbb{R}^N$ be a continuous function. Set

$$\tilde{P_N}(f) = P_N(f) - P_N(-f) \tag{2.4.14}$$

Thus $\tilde{P}_N : K \to \mathbb{R}^N$ is an odd continuous function. Given X_{N+1} an $(N+1)$-dimensional subspace of $\mathbb{X}$ and $\lambda > 0$ such that $\lambda S(X_{N+1}) \subseteq K$, then $\tilde{P}_N\Big|_{\partial(\lambda S(X_{N+1}))}$ is an odd continuous function from the boundary of an $(N+1)$-dimensional ball to $\mathbb{R}^N$. By Borsuk-Ulam theorem, there exists an $f^* \in \partial(\lambda S(X_{N+1}))$ (in particular $\|f\| = \lambda$) for which $\tilde{P}_N(f^*) = 0$. As a consequence, for any function $M_N : \mathbb{R}^N \to \mathbb{X}$

$$2f^* = [f^* - M_N(P_N(f^*))] - [-f^* - M_N(P_N(-f^*))] \tag{2.4.15}$$

and therefore

$$\max\{\|f^* - M_N(P_N(f^*))\|, \| - f^* - M_N(P_N(-f^*))\|\} \geq \|f^*\| = \lambda \tag{2.4.16}$$

Since both f^* and $-f^*$ are in K, this implies that $E(K; P_N, M_N; \mathbb{X}) \geq \lambda$. Since this inequality is valid for any choice of P_N and M_N and $\lambda \leq b_N(K; \mathbb{X})$, we have that $h_N(K; \mathbb{X}) \geq b_N(K; \mathbb{X})$ and the proof is done. ■

Lemma 2.4.4. *With the notation previously defined, there exists a constant C such that*

$$b_N(W_m^n; c) \geq CN^{-m/n} \tag{2.4.17}$$

PROOF.
Since b_N is a supremum, it suffices to prove that there exists a constant C and an $(N+1)$-dimensional linear space X_{N+1} such that $CN^{-m/n}S(X_{N+1}) \subseteq W_m^n$.

Let φ be any nonzero function in $\mathcal{C}^\infty(\mathbb{R}^n)$ with $\operatorname{supp}\varphi \subseteq [-1,1]^n$. For given n, m, we can choose φ satisfying $\|D^{\mathbf{k}}\varphi\| \leq 1$ for $|k| \leq m$. For $l > 0$ and $\mathbf{j} \in (2\mathbb{Z})^n$, set

$$\varphi_{\mathbf{j},l}(x_1, \ldots, x_n) = \varphi(x_1 l - j_1, \ldots, x_n l - j_n) \tag{2.4.18}$$

The support of $\varphi_{\mathbf{j},l}$ lies in $\prod_{i=1}^n [(j_i-1)/l, (j_i+1)/l]$. Since we are working with sup norm, we have that:

$$\|\varphi_{\mathbf{j},l}\| = \|\varphi\| \qquad \|D^{\mathbf{k}}\varphi_{\mathbf{j},l}\| = l^{|\mathbf{k}|}\|D^{\mathbf{k}}\varphi\| \tag{2.4.19}$$

For any fixed l, we observe that for different $\mathbf{j} \in (2\mathbb{Z})^n$, the supports of $\varphi_{\mathbf{j},l}$ are disjoint. Therefore, for any linear combination we have:

$$\left\|\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right\| = \|c\|_\infty\|\varphi\| \qquad \left\|D^{\mathbf{k}}\left(\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right)\right\| = l^{|\mathbf{k}|}\|c\|_\infty\|D^{\mathbf{k}}\varphi\| \tag{2.4.20}$$

where $\|c\|_\infty = \max_{\mathbf{j}} |c_{\mathbf{j}}|$.

$$\left\|\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right\|_{m,\infty} = \sum_{0\leq|\mathbf{k}|\leq m}\left\|D^{\mathbf{k}}\left(\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right)\right\| = \quad \text{(See definition 2.2.2)} \tag{2.4.21}$$

$$= \sum_{0\leq|\mathbf{k}|\leq m} l^{|\mathbf{k}|}\|c\|_\infty\|D^{\mathbf{k}}\varphi\| \leq \|c\|_\infty \sum_{0\leq|\mathbf{k}|\leq m} l^m = \quad (\|D^{\mathbf{k}}\varphi\| \leq 1 \text{ and } l^k \leq l^m) \tag{2.4.22}$$

$$= \|c\|_\infty\binom{n+m}{m}l^m = \binom{n+m}{m}l^m\left\|\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right\| \quad \text{The sum has } \tbinom{n+m}{n} \text{ terms.} \tag{2.4.23}$$

So for l large, the linear space generated by those $\varphi_{\mathbf{j},l}$ whose support lie totally in $[-1,1]^n$ is a linear space of dimension of the order of l^n with the property that

$$\left\|\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right\|_\infty \leq 1 \implies Cl^{-m}\left\|\sum_{\mathbf{j}} c_{\mathbf{j}}\varphi_{\mathbf{j},l}\right\|_{m,\infty} \leq 1 \tag{2.4.24}$$

for some constant C independent of l. This implies that $b_N(W_m^n; \mathcal{C}([-1,1]^n)) \geq Cl^{-m}$ where $N \approx l^n$. Thus we have proved the desired result

$$b_N(W_m^n; \mathcal{C}([-1,1]^n)) \geq CN^{-m/n} \tag{2.4.25}$$

■

2.4.1 Proof of lemma 2.4.2

Lemma 2.4.2 is a classical result whose proof is based on Jackson's Theorem (see theorem 2.4.5). It involves some concepts and previous results of approximation theory that will explained in this section.

2.4.1.1 Derivation from Jackson's Theorem

We begin by presenting the concepts of ***modulus of smoothness*** and ***best approximation*** by a polynomial.

Definition 2.4.4 (Modulus of smoothness). Given $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$, with partial derivatives up to order m, the ***modulus of smoothness of*** f ***of order*** m is a function $\omega_{f,m} : [0, \infty) \to [0, \infty)$ defined by

$$\omega_{f,m}(\delta) \stackrel{\text{def}}{=} \sup_{|\boldsymbol{\gamma}|=m} \left(\sup_{|\boldsymbol{x}-\boldsymbol{y}|\leq\delta} |D^{\boldsymbol{\gamma}} f(\boldsymbol{x}) - D^{\boldsymbol{\gamma}} f(\boldsymbol{y})| \right) \tag{2.4.26}$$

Definition 2.4.5 (Best approximation by polynomial). Given a function $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$, its best approximation error by a polynomial of degree k in the compact $K \subseteq \Omega$ is

$$E_k(f) \stackrel{\text{def}}{=} \inf_{p \in P_k^n} \|f - p\|_K \tag{2.4.27}$$

In this subsection we will always be using the sup norm for functions, and when it is not clear from the context, we will use the subindex to specify where the supremum is taken. So for example, if f is a function defined in K, $\|f\|_K = \sup_{x \in K} |f(x)|$.

Many results exist on upper bounds to this best approximation error. They are generally called Jackson's theorems (or Jackson's inequalities) in honor to D. Jackson, who first proved that type of results in [16].

The version of Jackson's theorem we will use is the following:

Theorem 2.4.5. *Let $f \in \mathcal{C}_0^m(\mathbb{R}^n)$ with $\operatorname{supp} f \subseteq K \subseteq \mathbb{R}^n$, and K compact, then*

$$E_k(f) \leq C \frac{1}{k^m} \omega_{f,m}\left(\frac{1}{k}\right) \tag{2.4.28}$$

where C is a constant depending only on n, m and K.

In our case $K = [-1, 1]^n$. Since $f \in \mathrm{W}_m^n$, in particular for any $\boldsymbol{\alpha} \in \mathbb{Z}_{\geq}^n$, $0 \leq |\boldsymbol{\alpha}| \leq m$, $\|D^{\boldsymbol{\alpha}} f\| \leq \sqrt{n}$ and therefore using triangular inequality, for any δ

$$\omega_{f,m}(\delta) \leq 2\sqrt{n} \tag{2.4.29}$$

Since C can depend on n, this $2\sqrt{n}$ factor can be added to the constant C, and we directly get the desired result.

2.4.1.2 Proof of Jackson's Theorem

In [6], the following result is proved:

Theorem 2.4.6. *Let $f \in \mathcal{C}_0^m(\mathbb{R}^n)$ with $\operatorname{supp} f \subseteq K \subseteq \mathbb{R}^n$, K compact, and $\boldsymbol{\alpha} \in \mathbb{Z}_{\geq 0}^n$, then there exists a polynomial p_k of degree at most k in n variables such that*

$$\|D^{\boldsymbol{\alpha}}(f - p_n)\|_K \leq C \frac{1}{k^{m-|\boldsymbol{\alpha}|}} \omega_{f,m-\boldsymbol{\alpha}}\left(\frac{1}{k}\right) \tag{2.4.30}$$

We are interested in the case $\boldsymbol{\alpha} = \mathbf{0}$. In the proof we will use some known concepts and results from the theory of Fourier transform.

Definition 2.4.6 (cf. Definition 7.1.2 in [15]). Let the set of functions $\mathscr{S}(\mathbb{R}^n)$ be defined by

$$\mathscr{S}(\mathbb{R}^n) = \{\varphi \in \mathcal{C}^\infty(\mathbb{R}^n) \,:\, \|x^{\boldsymbol{\beta}} D^{\boldsymbol{\alpha}} \varphi\| < \infty \quad \forall \boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{Z}_{\geq 0}^n\} \tag{2.4.31}$$

Note that $\mathcal{C}_0^\infty(\mathbb{R}^n) \subseteq \mathscr{S}(\mathbb{R}^n)$.

Notation. If $g : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ is a function and $\varepsilon > 0$, $g_{[\varepsilon]}$ is the function defined by

$$g_{[\varepsilon]}(\boldsymbol{x}) = \frac{1}{\varepsilon^n} g\left(\frac{\boldsymbol{x}}{\varepsilon}\right) \tag{2.4.32}$$

Also, if $\boldsymbol{z} \in \mathbb{C}^n$, we note its imaginary part as $\operatorname{Im}(\boldsymbol{z}) = (\operatorname{Im}(z_1), \ldots, \operatorname{Im}(z_n))$.

Lemma 2.4.7. *For any m, C, f, δ, the modulus of smoothness of f of order m has the following property:*

$$\omega_{f,m}(C\delta) \leq (C+1)\omega_{f,m}(\delta) \tag{2.4.33}$$

PROOF.
Given a multi-integer $\boldsymbol{\gamma}$,

$$\sup_{|\boldsymbol{x}-\boldsymbol{y}| \leq C\delta} |D^{\boldsymbol{\gamma}} f(\boldsymbol{x}) - D^{\boldsymbol{\gamma}} f(\boldsymbol{y})| = \sup_{|\boldsymbol{x}-\boldsymbol{y}| \leq \delta} |D^{\boldsymbol{\gamma}} f(\boldsymbol{x}C) - D^{\boldsymbol{\gamma}} f(\boldsymbol{y}C)| \tag{2.4.34}$$

Now the idea is to consider the interval that goes from $\boldsymbol{x}$ to $\boldsymbol{y}$, that are the points $t\boldsymbol{x} + (1-t)\boldsymbol{y}$, for $t \in [0,1]$, and apply triangular inequality for points in that interval. If C is integer, then we can introduce $\sum_{j=1}^{C-1} D^{\boldsymbol{\gamma}} f(C\boldsymbol{x} + (C-j)\boldsymbol{y})$ and apply C times triangular inequality to get

$$|D^{\boldsymbol{\gamma}} f(\boldsymbol{x}C) - D^{\boldsymbol{\gamma}} f(\boldsymbol{y}C)| \leq \sum_{j=0}^{C-1} \left| D^{\boldsymbol{\gamma}} f\Big(j\boldsymbol{x} + (C-j)\boldsymbol{y}\Big) - D^{\boldsymbol{\gamma}} f\Big((j+1)\boldsymbol{x} + (C-(j+1))\boldsymbol{y}\Big) \right| \tag{2.4.35}$$

Since for every j, $|j\boldsymbol{x} + (C-j)\boldsymbol{y} - (j+1)\boldsymbol{x} - [C-(j+1)]\boldsymbol{y}| = |\boldsymbol{x}-\boldsymbol{y}|$, and recalling eq. (2.4.34), for C integer we have:

$$\omega_{f,m}(C\delta) \leq C\omega_{f,m}(\delta) \tag{2.4.36}$$

Now if C is not integer, let $\lceil C \rceil$ be the smallest integer which is greater than or equal to C. It is trivially satified:

$$\omega_{f,m}(C\delta) \leq \omega_{f,m}(\lceil C \rceil \delta) \leq \lceil C \rceil \omega_{f,m}(\delta) \leq (C+1)\omega_{f,m}(\delta) \tag{2.4.37}$$

■

Lemma 2.4.8. *Let r be a nonnegative integer and $f \in \mathcal{C}_0^r(\mathbb{R}^n)$. For any pair of points $\boldsymbol{x}, \boldsymbol{h} \in \mathbb{R}^n$, the quantity $R(\boldsymbol{x}, \boldsymbol{h})$ defined by*

$$f(\boldsymbol{x}+\boldsymbol{h}) = \sum_{0 \leq |\boldsymbol{\alpha}| \leq r} \frac{D^{\boldsymbol{\alpha}} f}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}} + R(\boldsymbol{x}, \boldsymbol{h}) \tag{2.4.38}$$

where $\boldsymbol{\alpha}! = \prod_{i=1}^n \alpha_i!$ and $\boldsymbol{h}^{\boldsymbol{\alpha}} = \prod_{i=1}^n h_i^{\alpha_i}$, satisfies

$$|R(\boldsymbol{x}, \boldsymbol{h})| \leq \frac{n^r |\boldsymbol{h}|^r \omega_{f,r}(|\boldsymbol{h}|)}{r!} \tag{2.4.39}$$

PROOF.
Let us define the function $u : \mathbb{R} \to \mathbb{R}$ by $u(t) \stackrel{\text{def}}{=} f(\boldsymbol{x} + t\boldsymbol{h})$. Applying chain rule for derivatives, the l-th derivative of u is

$$u^{(l}(t) = l! \sum_{|\boldsymbol{\alpha}|=l} \frac{D^{\boldsymbol{\alpha}} f(\boldsymbol{x}+t\boldsymbol{h})}{\boldsymbol{\alpha}!} \boldsymbol{h}^{\boldsymbol{\alpha}} \tag{2.4.40}$$

In addition:

$$\omega_{u,r}(\delta) = \sup_{|t_1 - t_2| \leq \delta} \left| u^{(r}(t_1) - u^{(r}(t_2) \right| \tag{2.4.41}$$

$$= \sup_{|t_1 - t_2| \leq \delta} r! \sum_{|\boldsymbol{\alpha}|=k} \left| \frac{\boldsymbol{h}^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} [D^{\boldsymbol{\alpha}} f(\boldsymbol{x} + t_1\boldsymbol{h}) - D^{\boldsymbol{\alpha}} f(\boldsymbol{x} + t_2\boldsymbol{h})] \right| \qquad \text{By eq. (2.4.40)} \tag{2.4.42}$$

$$\tag{2.4.43}$$

Each term in the sum of eq. (2.4.42) is less or equal than the supremum for the first inequality and the supremum is taken only in one of the directions in $\omega_{f,r}$ for the second one we get:

$$\omega_{u,r}(\delta) \leq r!|\boldsymbol{h}|^r \left(\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{\boldsymbol{\alpha}!} \right) \sup_{|t_1\boldsymbol{h} - t_2\boldsymbol{h}| \leq \delta |\boldsymbol{h}|} \sup_{|\boldsymbol{\alpha}|=k} |D^{\boldsymbol{\alpha}} f(\boldsymbol{x} + t_1\boldsymbol{h}) - D^{\boldsymbol{\alpha}} f(\boldsymbol{x} + t_2\boldsymbol{h})| \tag{2.4.44}$$

$$\leq r!|\boldsymbol{h}|^r \left(\sum_{|\boldsymbol{\alpha}|=k} \frac{1}{\boldsymbol{\alpha}!} \right) \omega_{f,r}(\delta|h|) \tag{2.4.45}$$

$$\tag{2.4.46}$$

And finally, using $\sum_{|\boldsymbol{\alpha}|=r} \frac{r!}{\boldsymbol{\alpha}!} = r^n$ we have that

$$\omega_{u,r}(\delta) \leq n^r |\boldsymbol{h}|^r \omega_{f,r}(\delta|\boldsymbol{h}|) \tag{2.4.47}$$

Observe that

$$R(\boldsymbol{x},\boldsymbol{h}) = f(\boldsymbol{x}+\boldsymbol{h}) - \sum_{0\leq|\boldsymbol{\alpha}|\leq r} \frac{1}{|\boldsymbol{\alpha}|!}|\boldsymbol{\alpha}|! \frac{D^{\boldsymbol{\alpha}} f(\boldsymbol{x})}{\boldsymbol{\alpha}!}\boldsymbol{h}^{\boldsymbol{\alpha}} = u(1) - \sum_{l=1}^{r} \frac{u^{(l}(0)}{l!} \tag{2.4.48}$$

Taylor's theorem with the mean-value form of the reminder states that there exists a $\xi \in [0,1]$ such that $u(1) - \sum_{l=1}^{r-1} \frac{u^{(l}(0)}{l!} = \frac{u^{(r}(\xi)}{r!}$. Applying this and eq. (2.4.47) with $\delta = 1$ we finish the proof:

$$|R(\boldsymbol{x},\boldsymbol{h})| = \left| \frac{u^{(r}(\xi)}{r!} - \frac{u^{(r}(0)}{r!} \right| \leq \frac{\omega_{u,r}(1)}{r!} \leq \frac{n^r |\boldsymbol{h}|^r \omega_{f,r}(|\boldsymbol{h}|)}{r!} \tag{2.4.49}$$

■

Lemma 2.4.9. *Let δ be a fixed positive constant. Then there exists an holomorphic function $G : \mathbb{C}^n \to \mathbb{C}$ and a positive constant A satisfying*

$$|G(\boldsymbol{z})| \leq A e^{\delta|\mathrm{Im}(\boldsymbol{z})|} \quad \forall \boldsymbol{z} \in \mathbb{C}^n \tag{2.4.50}$$

Such that the restriction $g = G\big|_{\mathbb{R}^n}$ satisfies:

a) $g \in \mathscr{S}(\mathbb{R}^n)$

b) For any integer $r \geq 0$, let

$$I_r = \frac{n^r}{r!} \int_{\mathbb{R}^n} |\boldsymbol{w}|^r (|\boldsymbol{w}|+1)|g(\boldsymbol{w})| d\boldsymbol{w} \tag{2.4.51}$$

then for all $f \in \mathcal{C}_0^r(\mathbb{R}^n)$ and $\varepsilon > 0$ it is satisfied that

$$\|f - g_{[\varepsilon]} * f\| \leq I_r \varepsilon^r \omega_{f,r}(\varepsilon) \tag{2.4.52}$$

PROOF.
Let $\Phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$ such that

(*i*) $0 \leq \Phi \leq 1$

(*ii*) There exists an open neighborhood of 0 such that $\Phi = 1$ in it.

(*iii*) $\mathrm{supp}\,\Phi \subseteq \overline{B(\mathbf{0},\delta)}$ (closed ball of radius δ around the origin)

We define $G(\boldsymbol{z})$ as a Fourier transform of Φ:

$$G(\boldsymbol{z}) \stackrel{\text{def}}{=} \frac{1}{(2\pi)^n} \int_{\mathbb{R}^n} \Phi(\boldsymbol{\xi}) e^{-i\boldsymbol{\xi}\cdot\boldsymbol{z}} d\boldsymbol{\xi} \qquad \boldsymbol{z} \in \mathbb{C}^n \tag{2.4.53}$$

where $\boldsymbol{x} \cdot \xi$ is the ordinary scalar product. Since Φ has compact support, G is well defined for all $\boldsymbol{z} \in \mathbb{C}^n$. Moreover, the smoothness of Φ lets us exchange derivatives by integrals and gives G is holomorphic in all $\mathbb{C}^n$. Since $\boldsymbol{\xi}$ is real, we can do the following decomposition:

$$e^{-i\boldsymbol{\xi}\cdot\boldsymbol{z}} = e^{-i\boldsymbol{\xi}(\mathrm{Re}(\boldsymbol{z})+i\mathrm{Im}(\boldsymbol{z}))} = e^{\boldsymbol{\xi}\cdot\mathrm{Im}(\boldsymbol{z})}e^{-i\boldsymbol{\xi}\cdot\mathrm{Re}(\boldsymbol{z})} \tag{2.4.54}$$

where the second factor has modulus 1. Using this decomposition and the triangular inequality we get:

$$|G(\boldsymbol{z})| \leq \frac{1}{(2\pi)^n}\int_{\mathbb{R}^n} \Phi(\boldsymbol{\xi})e^{\boldsymbol{\xi}\cdot\mathrm{Im}(\boldsymbol{z})}d\boldsymbol{\xi} \tag{2.4.55}$$

By property (iii) of Φ, eq. (2.4.50) is satisfied with $A = \frac{1}{(2\pi)^n}\int_{\mathbb{R}^n}\Phi(\boldsymbol{\xi})d\boldsymbol{\xi}$. Since $\Phi \in \mathcal{C}_0^\infty(\mathbb{R}^n)$, its Fourier transform is also in $\mathscr{S}(\mathbb{R}^n)$ (this implies $g \in \mathscr{S}(\mathbb{R}^n)$), and applying the Fourier inversion formula [15, Th. 7.1.5]:

$$\Phi(\boldsymbol{\xi}) = \int_{\mathbb{R}^n} g(\boldsymbol{x})e^{i\boldsymbol{x}\cdot\boldsymbol{\xi}}d\boldsymbol{x} \qquad \boldsymbol{\xi} \in \mathbb{R}^n \tag{2.4.56}$$

Note that by setting $\boldsymbol{\xi} = 0$ in eq. (2.4.56) and using (ii) we get that

$$\int_{\mathbb{R}^n} g(\boldsymbol{x})d\boldsymbol{x} = 1 \tag{2.4.57}$$

If previously we differentiate in eq. (2.4.56) with respect to the multi-integer $\boldsymbol{j} = (j_1, \ldots, j_n) \in \mathbb{Z}^n_{\geq 0}$, $\boldsymbol{j} \neq \boldsymbol{0}$, we get

$$\int_{\mathbb{R}^n} x_1^{j_1}\cdots x_n^{j_n} g(\boldsymbol{x})d\boldsymbol{x} = 0 \tag{2.4.58}$$

We now move to prove property (b). For $\boldsymbol{x} \in \mathbb{R}^n$ and $\varepsilon > 0$:

$$(g_{[\varepsilon]} * f - f)(\boldsymbol{x}) = \frac{1}{\varepsilon^n}\int f(\boldsymbol{x}-\boldsymbol{w})g\left(\frac{\boldsymbol{w}}{\varepsilon}\right)d\boldsymbol{w} - f(\boldsymbol{x}) \tag{2.4.59}$$

Applying a change of variables $\boldsymbol{y} = \boldsymbol{w}/\varepsilon$, the first term becomes $\int_{\mathbb{R}^n} f(\boldsymbol{x}-\varepsilon\boldsymbol{y})g(\boldsymbol{y})d\boldsymbol{y}$ and applying eq. (2.4.57), so $f(\boldsymbol{x}) = \int_{\mathbb{R}^n} f(\boldsymbol{x})g(\boldsymbol{w})d\boldsymbol{w}$ we get

$$(g_{[\varepsilon]} * f - f)(\boldsymbol{x}) = \int_{\mathbb{R}^n}[f(\boldsymbol{x}-\varepsilon\boldsymbol{w}) - f(\boldsymbol{x})]g(\boldsymbol{w})d\boldsymbol{w} \tag{2.4.60}$$

$$= \int_{\mathbb{R}^n} R(\boldsymbol{x}, \varepsilon\boldsymbol{w})g(\boldsymbol{w})d\boldsymbol{w} \qquad \text{Definition of } R(\boldsymbol{x},\boldsymbol{h}) \text{ and eq. (2.4.58)} \tag{2.4.61}$$

Now using lemma 2.4.8 and lemma 2.4.7 we have that

$$|R(\boldsymbol{x}, \varepsilon\boldsymbol{w})| \leq \frac{n^r\varepsilon^r}{r!}\omega_{f,r}(\varepsilon)|\boldsymbol{w}|^r(|\boldsymbol{w}|+1) \tag{2.4.62}$$

from which it directly follows

$$\|g_{[\varepsilon]} * f - f\| \leq \frac{n^r\varepsilon^r}{r!}\omega_{f,r}(\varepsilon)\int_{\mathbb{R}^n}|\boldsymbol{w}^r|(|\boldsymbol{w}|+1)|g(\boldsymbol{w})|d\boldsymbol{w} \tag{2.4.63}$$

which finishes the proof. ■

Definition 2.4.7 (McLaurin polynomials). For any $\boldsymbol{\alpha} \in \mathbb{Z}^n_{\geq 0}$, let $a_{\boldsymbol{\alpha}}$ be its corresponding coefficient on its McLaurin series, so if f is holomorphic in an open neighborhood of 0 fulfills $f(\boldsymbol{z}) = \sum_{\boldsymbol{\alpha}} a_{\boldsymbol{\alpha}} \boldsymbol{z}^{\boldsymbol{\alpha}}$. Then for any nonnegative integer k, the k-th McLaurin polynomial of f is defined as

$$p_{f,k} = \sum_{0 \leq |\boldsymbol{\alpha}| \leq k} a_{\boldsymbol{\alpha}} \boldsymbol{z}^{\boldsymbol{\alpha}} \tag{2.4.64}$$

Let $R \geq 0$, we define E_R as the disk of radius R in $\mathbb{C}^n$, namely $E_R = \{\boldsymbol{z} \in \mathbb{C}^n \, : \, |z_i| \leq R \quad \forall i\}$

Lemma 2.4.10. *Let $0 < R < S$. Let f be an holomorphic function in an open neighborhood of E_S satisfying $\|f\|_{E_S} \leq M$. Then*

$$\|f - p_{f,k}\|_{E_R} \leq \frac{M}{1 - R/S} \left(\frac{R}{S}\right)^{k+1} \tag{2.4.65}$$

PROOF.
We first prove it for $n = 1$. In that case, let $f(z) = \sum_{\alpha=0}^{\infty} a_\alpha z^\alpha$. Using the definition of a_α and the standard bound for the integral, the hypothesis $\|f\|_{E_S} \leq M$ implies

$$|a_\alpha| = \left| \frac{1}{2\pi i} \oint_{|z|=S} \frac{f(z)}{z^{\alpha+1}} dz \right| \leq \frac{1}{2\pi} \cdot (2\pi S) \frac{M}{S^{\alpha+1}} = \frac{M}{S^\alpha} \tag{2.4.66}$$

Now we observe that $(f - p_{f,k})(z) = \sum_{\alpha=k+1}^{\infty} a_\alpha z^\alpha$. Combining both results:

$$\|f - p_{f,k}\|_{E_R} = \sup_{|z| \leq R} \left| \sum_{\alpha=k+1}^{\infty} a_\alpha z^\alpha \right| \tag{2.4.67}$$

$$\leq \sup_{|z| \leq R} \sum_{\alpha=k+1}^{\infty} |a_\alpha| |z^\alpha| \qquad \text{Triangular inequality} \tag{2.4.68}$$

$$\leq \sum_{\alpha=k+1}^{\infty} \frac{M}{S^\alpha} R^\alpha \qquad \text{eq. (2.4.66) and } |z| \leq R \tag{2.4.69}$$

$$= \frac{M}{1 - R/S} \left(\frac{R}{S}\right)^{k+1} \qquad \text{Geometric series formula} \tag{2.4.70}$$

as required.

For the case of n general, consider a fixed point $\boldsymbol{Z} = (Z_1, \ldots, Z_n) \in \mathbb{C}^n$ such that $|Z_j| \leq 1$ for $j = 1 : n$. Let $\eta : \mathbb{C} \to \mathbb{C}$ be defined by $\eta(\lambda) = f(\lambda \boldsymbol{Z})$. By the chain rule, the k-th derivative of η is

$$\eta^{(k}(\lambda) = \sum_{0 \leq |\boldsymbol{\alpha}| \leq k} \boldsymbol{Z}^{\boldsymbol{\alpha}} D^{\boldsymbol{\alpha}} f(\lambda \boldsymbol{Z}) \tag{2.4.71}$$

and because $|Z_j| \leq 1$, the k-th McLaurin coefficient of η is

$$a_k^\eta = \frac{1}{2\pi i} \oint_{|z|=S} \frac{\eta^{(k}(\lambda)}{z^{k+1}} \tag{2.4.72}$$

and from this it can be derived that $p_{\eta,k}(\lambda) = p_{f,k}(\lambda \boldsymbol{Z})$, so the result for $n = 1$ can be applied. ■

Corollary 2.4.11. *Let $R > 0$, $S > R+1$ and f be an holomorphic function in an open neighborhood of E_S such that $\|f\|_{E_S} \leq M$. Then*

$$\|f - p_{f,k}\|_{E_R} \leq \frac{M}{1 - R/(S-1)} \left(\frac{R}{S-1}\right)^{k+1} \tag{2.4.73}$$

PROOF.
If $S > R + 1$, then $S - 1 > R$, so we can apply lemma 2.4.10 to $S - 1$ and R instead of S and R, and directly get the result. ■

Lemma 2.4.12. *Let $R > 0$ and $f \in \mathcal{C}^m(\mathbb{R}^n)$ with $\operatorname{supp} f \subseteq [-R, R]^n$. Then for all positive integer k the following inequality holds:*

$$\|f\| \leq R^m (kR + 1)\omega_{f,m}\left(\frac{1}{k}\right) \tag{2.4.74}$$

PROOF.
First we prove that

$$\|f\| \leq R^m \sup_{|\boldsymbol{\alpha}|=m} \|D^\alpha f\| \tag{2.4.75}$$

The case $m = 0$ is obvious. For the case $m = 1$, for each $r \in [-R, R]$, consider the intervals I_r of length R as the intervals $[r - R, r] \times \{0\}^{n-1}$ for $r \in [-R, 0]$ (see fig. 2.3). Integrating $\partial_{x_1} f$ along I_r we get:

$$\int_{I_r} \partial_{x_1} f = \operatorname{sgn}(r) f(r) \qquad \left|\int_{I_r} \partial_{x_1} f\right| \leq R \sup_{[r-R,r]\times[-R,R]^{n-1}} |\partial_{x_1} f| \tag{2.4.76}$$

Making an analogous construction for intervals I_r defined as $[r, r+R] \times \{0\}^{n-1}$ for $r \in [0, R]$, and the analogous construction for the rest of the partial derivatives $\partial_{x_j} f$ for $j = 2 : n$ we get the case $m = 1$. The case for $m > 1$ is proved by applying the case $m = 1$ repeatedly.

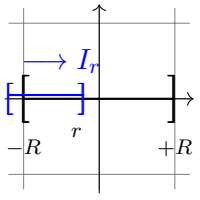


Figure 2.3: Partial sketch of the proof.

Now by the definition of modulus of smoothness, since $\operatorname{supp} f \subseteq [-R, R]^n$, it is clear that

$$\sup_{|\alpha|=m} \|D^\alpha f\| \leq \omega_{f,m}(R) = \omega_{f,m}\left(k \cdot \frac{R}{k}\right) \leq (kR+1)\omega_{f,m}\left(\frac{1}{k}\right) \tag{2.4.77}$$

where the inequality comes from applying lemma 2.4.7. Equation (2.4.75) and eq. (2.4.77) together yield to the desired result. ■

PROOF. (Of theorem 2.4.6)
Let $f \in \mathcal{C}_0^m(\mathbb{R}^n)$. Let R be the diameter of K. Wlog (by a suitable translation in $\mathbb{R}^n$) we can assume $K \subseteq [-R, R]^n$. Let $\delta > 0$ be a fixed real number such that

$$\sqrt{n}(2R+1)\delta < \ln 2 \tag{2.4.78}$$

where $\ln 2$ is the natural logarithm of 2.

Let G and $g = G|_{\mathbb{R}^n}$ be the functions of lemma 2.4.9 associated with δ and for any integer $r \geq 0$, let I_r be as in lemma 2.4.9. From eq. (2.4.78) we see that there exists a constant k_0 such that for all $k \geq k_0$

$$(kR+1)(2Rk)^n e^{\sqrt{n}(2R+1)\delta k} 2^{-k} \leq \frac{I_r}{k^r} \tag{2.4.79}$$

Therefore there exists a constant $C \geq I_r$ such that for all $k \geq 1$

$$(kR+1)(2Rk)^n e^{\sqrt{n}(2R+1)\delta k} 2^{-k} \leq \frac{C}{k^r} \tag{2.4.80}$$

For the rest of the proof, k will be a fixed positive integer.

Let $H : \mathbb{C}^n \to \mathbb{C}$ be the function such $g_{\lfloor \frac{1}{k} \rfloor} * f = H|_{\mathbb{R}^n}$,

$$H(\boldsymbol{z}) \stackrel{\text{def}}{=} k^n \int_{\mathbb{R}^n} G\big(k(\boldsymbol{z} - \boldsymbol{w})\big) f(\boldsymbol{w}) d\boldsymbol{w} \qquad \boldsymbol{z} \in \mathbb{C}^n \tag{2.4.81}$$

From the above results it follows:

$$\sup_{E_{2R+1}} |H(\boldsymbol{z})| \leq k^n e^{\sqrt{n}(2R+1)\delta k} \int_{\mathbb{R}^n} |f(\boldsymbol{w})| s d\boldsymbol{w} \qquad \text{lemma 2.4.9} \tag{2.4.82}$$

$$\leq A(2Rk)^n R^m (kR+1) e^{\sqrt{n}(2R+1)\delta k} \omega_{f,m}\left(\frac{1}{k}\right) \qquad \text{lemma 2.4.12} \tag{2.4.83}$$

From lemma 2.4.9 it follows:

$$\sup_{\mathbb{R}^n} |f - g_{\lfloor \frac{1}{k} \rfloor} * f| \leq \frac{C}{k^m} \omega_{f,m}\left(\frac{1}{k}\right) \tag{2.4.84}$$

And from lemma 2.4.10 with $S = R + 1$ it follows

$$\sup_{[-R,R]^n} |H - p_{H,k}| \leq A R^m (kR+1)(2Rk)^n e^{\sqrt{n}(2R+1)\delta k} 2^{-k} w_{f,m}\left(\frac{1}{k}\right) \tag{2.4.85}$$

Now the result comes directly from the concatenation of previous inequalities eq. (2.4.85), eq. (2.4.84) and eq. (2.4.80). ■

Chapter 3

Universality Theorems

3.1 Framework

A fundamental question that has been answered about neural networks is its representation potential: are they able to approximate any function to any precision?

The proof depends on the graph $\mathcal{G}$ associated with the network and the activation function σ. The main result of this chapter, obtained in [24, Th. 2.1] is that a shallow network with activation function satisfying certain week hypothesis (the result even works for many functions with essential discontinuities) are universal if and only if the activation function σ is not a polynomial. This is a very powerful result and its complete proof uses advanced analysis, so we will prove the particular case of σ being continuous, which we consider illustrative enough for our purpose, and a wide enough result, since most learning algorithms use continuous activation functions.

Definition 3.1.1 (Density property). Let $\mathbb{X}$ be a space of functions with some domain $\Omega \subseteq \mathbb{R}^n$, and let $\mathcal{F} \subseteq \mathbb{X}$ be a family of functions in this space. We say that $\mathcal{F}$ is ***universal*** or ***dense*** if for every $g \in \mathbb{X}$ and for every compact $K \subseteq \Omega$, there exists a sequence of functions $\{f_i\}_{i\in\mathbb{N}} \subseteq \mathcal{F}$, such that

$$\lim_{i\to\infty} \|g - f_i\|_{\mathcal{L}^\infty(K)} = 0 \tag{3.1.1}$$

We have defined density with the *sup* norm. An analogous definition can be stated with any other p-norm, but we have chosen to fix the norm for the sake of simplicity.

In particular, latter in the chapter we will comment how our theorems generalize to p-norms.

3.2 Universality of Shallow Networks

In this section we will prove that shallow networks with arbitrary high number of units and a non polynomial continuous activation function are universal.

The proof is not straightforward and will require some definitions and previous results. The objective is to prove the following result:

Theorem 3.2.1 (Universality theorem for $\sigma \in \mathcal{C}(\mathbb{R})$). *Shallow networks with an arbitrary number of units and activation function $\sigma \in \mathcal{C}(\mathbb{R})$ are universal in $\mathcal{C}(\mathbb{R}^n) \iff \sigma$ is not a polynomial.*

The whole proof is a bit tedious, specially for those readers not used to mathematical analysis. We have marked with an asterisk (*) those proofs that we consider of special interest, either because of the relevance of the result or the ideas leading to a given result. In particular, some ideas exposed in proofs marked with an asterisk, will be used in following chapters.

Let $\boldsymbol{x} = (x_1, \dots, x_n)$, $\boldsymbol{m} = (m_1, \dots, m_n)$. be vectors in $\mathbb{R}^n$. We use the following notation:

$$\boldsymbol{x}^{\boldsymbol{m}} = x_1^{m_1} \cdots x_n^{m_n} \qquad ; \qquad |\boldsymbol{m}| = m_1 + \cdots + m_n \tag{3.2.1}$$

Let H_k^n be the set of homogeneous polynomials of n coordinates and degree k and P_k^n be the set of polynomials of degree at most k (for P_k^n we include the case $k = \infty$). I.e.:

$$H_k^n = \left\{ \sum_{|\boldsymbol{m}|=k} a_{\boldsymbol{m}} \boldsymbol{x}^{\boldsymbol{m}} \right\} \qquad P_k^n = \bigcup_{i=0}^{k} H_i^n \tag{3.2.2}$$

Note that H_k^n is a vector space of dimension $\binom{n-1+k}{k}$

The main result of this section is theorem 3.2.5 and its corollaries. To arrive there, we will need some previous results:

Lemma 3.2.2. *Let $\Omega \subseteq \mathbb{R}^n$, let $L(\Omega) = \bigcup_{\boldsymbol{a} \in \Omega} \operatorname{span}\{\boldsymbol{a}\}$. If the only polynomial in P_∞^n that vanishes in $L(\Omega)$ is the trivial one, then the set*

$$\mathscr{M}(\Omega) = \operatorname{span}\{g(\boldsymbol{a} \cdot \boldsymbol{x}) \ : \ \boldsymbol{a} \in \Omega,\ g \in \mathcal{C}(\mathbb{R})\} \tag{3.2.3}$$

is dense in $\mathcal{C}(\mathbb{R}^n)$.

PROOF. (*)
We will prove that under the stated hypothesis, for any k, $H_k^n \subseteq \mathscr{M}(\Omega)$. A generalized version of this proof can be found in [25, Theorem 2.1]. Applying Stone-Weierstrass's theorem [21, Ch. 3] one can complete the proof.

First we prove that for any $\boldsymbol{d} \in L(\Omega)$, $(\boldsymbol{d}\cdot\boldsymbol{x})^k \in \mathscr{M}(\Omega)$.

To do so, consider $\boldsymbol{d} \in L(\Omega)$, by the definition of $L(\Omega)$, $\boldsymbol{d} \in \text{span}(\boldsymbol{a})$ for some $\boldsymbol{a} \in \Omega$, so there exists $y \in \mathbb{R}$ such that $\boldsymbol{d} = y\boldsymbol{a}$. So the function $(\boldsymbol{d}\cdot\boldsymbol{x})^k$ can be written as: $(\boldsymbol{d}\cdot\boldsymbol{x})^k = (y\boldsymbol{a}\boldsymbol{x})^k = g(\boldsymbol{x}\cdot\boldsymbol{a}) \in \mathscr{M}(\Omega)$, with $g(x) = (yx)^k$.

Now consider the dual space of H_k^n, defined as

$$H_k^{n*} = \{\sigma : H_k^n \to \mathbb{R} \ : \ \sigma \text{ is a linear function}\} \tag{3.2.4}$$

Since H_k^n is a finite vector space, H_k^{n*} is a finite vector space of the same dimension. A basis of this space is $V = \{D^{\boldsymbol{m}} \ : \ |\boldsymbol{m}| = k\}$ where

$$\begin{aligned} D : H_k^n &\to \mathbb{R} \\ p(\boldsymbol{x}) &\mapsto D^{\boldsymbol{m}}p(\boldsymbol{x}) = \frac{\partial^{|\boldsymbol{m}|}p(\boldsymbol{x})}{\partial x_1^{m_1}\cdots\partial x_n^{m_n}} \end{aligned}$$

It is a basis because it has the right number of elements and they are mutually independent since:

$$D^{\boldsymbol{m}}\boldsymbol{x}^{\boldsymbol{m'}} = \begin{cases} 0 & \text{if } \boldsymbol{m} \neq \boldsymbol{m'} \\ m_1!\cdots m_n! & \text{if } \boldsymbol{m} = \boldsymbol{m'} \end{cases} \tag{3.2.5}$$

Since V is a basis of H_k^{n*}, any element can be written as a linear combination of elements in V. Equivalently, any linear function that maps H_k^n to $\mathbb{R}$ can be written in terms of a polynomial $q \in H_k^n$ as

$$\begin{aligned} f_q : H_k^n &\to \mathbb{R} \\ p &\mapsto q(D)p \end{aligned}$$

We want to study how this functions f_q act on functions of the form $(\boldsymbol{d}\cdot\boldsymbol{x})^k$. We first study the case of q being a monomial, i.e. $q(\boldsymbol{x}) = \boldsymbol{x}^{\boldsymbol{m}}$ for some $\boldsymbol{m} \in \mathbb{Z}_+^n$, $|\boldsymbol{m}| = k$. Applying derivative properties:

$$q(D)(\boldsymbol{d}\cdot\boldsymbol{x})^k \ = \ D^{\boldsymbol{m}}(d_1x_1 + \cdots + d_nx_n)^k \ = \ kd_1D^{(m_1-1,\ldots,m_n)}(\boldsymbol{d}\cdot\boldsymbol{x})^{k-1} \ = \ \cdots \ = \ k!q(\boldsymbol{d}) \tag{3.2.6}$$

Using linearity, the previous result generalizes to all $q \in H_k^n$. With this result, we want to study what happens when we apply f_q functions to the linear subspace of polynomials

$$W = \text{span}\{(\boldsymbol{d}\cdot\boldsymbol{x})^k \in : \ d \in L(\Omega)\} \subseteq H_k^n \tag{3.2.7}$$

If some $f_q \in H_k^{n*}$ annihilates all polynomials in W, in particular it annihilates all polynomials of the form $(\boldsymbol{d}\cdot\boldsymbol{x})^k$ for all $\boldsymbol{d} \in L(\Omega)$. Since $f_q(\boldsymbol{d}\cdot\boldsymbol{x})^k = k!q(\boldsymbol{d})$, this means that q vanishes in $L(\Omega)$. By hypothesis, $q = 0$ (as a polynomial). In terms of linear algebra, W is a subspace of H_k^n with the property that any linear function $\sigma : H_n^k \to \mathbb{R}$ is trivial if and only if the restriction $\sigma\big|_W : W \to \mathbb{R}$ is trivial. This implies $W = H_k^n$. The proof is finished observing $H_k^n \subseteq W \subseteq \mathscr{M}(\Omega)$. ■

The next corollary follows immediately from the proof of this lemma and will be useful for another result (theorem 4.2.1).

Corollary 3.2.3. *Let $r = dim\ H_k^n = \binom{n-1+k}{k}$ and $s = dim\ P_k^n = \binom{n+k}{k}$. Then there exist $\{\boldsymbol{a}^i\}_{i=1:r} \subseteq \mathbb{R}^n$, $\{\boldsymbol{b}^i\}_{i=1:s} \subseteq \mathbb{R}^n$, $\{f_i\}_{i=1:r} \subseteq H_k^n$ and $\{g_i\}_{i=1:s} \subseteq P_k^n$ such that*

$$H_k^n = \left\{ \sum_{i=1}^{r} f_i(\boldsymbol{a}^i \cdot \boldsymbol{x}) \ : \ f_i \in H_k^1 \right\} \qquad P_k^n = \left\{ \sum_{i=1}^{s} g_i(\boldsymbol{b}^i \cdot \boldsymbol{x}) \ : \ g_i \in P_k^1 \right\} \tag{3.2.8}$$

PROOF. (*)
Consider the case $\Omega = \mathbb{R}^n$ of the previous lemma. For each j, we have that

$$H_j^n = \text{span}\{(\boldsymbol{d} \cdot \boldsymbol{x})^j \in : \ d \in \mathbb{R}^n\} \tag{3.2.9}$$

As a consequence, there exist $\{\boldsymbol{a}^i\}_{i=1:r} \subseteq \mathbb{R}^n$ such that $\{(\boldsymbol{a}^i \cdot \boldsymbol{x})^j\}_{i=1:r}$ is a basis of H_j^n. This is equivalent to

$$H_j^n = \left\{ \sum_{i=1}^{r} g_i(\boldsymbol{a}^i \cdot \boldsymbol{x}) \ : \ g_i \in H_j^1 \ \forall i \right\} \tag{3.2.10}$$

The analogous version for non homogeneous polynomials follows directly because $P_k^n = \bigcup_{j=1}^{k} H_j^n$ ■

Lemma 3.2.4. *If $\mathcal{S}_1(\sigma, \mathbb{R})$ is dense in $\mathcal{C}(\mathbb{R})$, then $\mathcal{S}_n(\sigma, \mathbb{R}^n)$ is dense in $\mathcal{C}(\mathbb{R}^n)$.*

PROOF.
Let $g \in \mathcal{C}(\mathbb{R}^n)$ and $K \subseteq \mathbb{R}^n$ compact. Lemma 3.2.2 states that $\mathscr{M} = \text{span}\{f(\boldsymbol{a} \cdot \boldsymbol{x}) \ : \ \boldsymbol{a} \in \mathbb{R}^n, \ f \in \mathcal{C}(\mathbb{R})\}$ is dense in $\mathcal{C}(K)$. Thus given $\varepsilon > 0$, there exists $k \in \mathbb{N}$ and a sequence of functions $\{f_i\}_{i=1:k} \subseteq \mathcal{C}(\mathbb{R})$ and a sequence of vectors $\{\boldsymbol{a}^i\}_{i=1:k} \subseteq \mathbb{R}^n$ such that for all $\boldsymbol{x} \in K$:

$$\left| g(\boldsymbol{x}) - \sum_{i=1}^{k} f_i(\boldsymbol{a}^i \cdot \boldsymbol{x}) \right| < \frac{\varepsilon}{2} \tag{3.2.11}$$

Since K is compact, for all $i = 1 : k$ there exists a finite interval $[\alpha_i, \beta_i]$ such that

$$\{\boldsymbol{a}^i \cdot \boldsymbol{x} \ : \ \boldsymbol{x} \in K\} \subseteq [\alpha_i, \beta_i] \tag{3.2.12}$$

Because $\mathcal{S}_1$ is dense in $[\alpha_i, \beta_i]$, there exists $m_1 \in \mathbb{N}$ and constants $c_{ij}, w_{ij}, \theta_{ij}$ such that for all $y \in [\alpha_i, \beta_i]$:

$$\left| f_i(y) - \sum_{j=1}^{m_i} c_{ij} \sigma\left(w_{ij} y + \theta_{ij}\right) \right| < \frac{\varepsilon}{2k} \tag{3.2.13}$$

Applying both inequalities yields, for all $\boldsymbol{x} \in K$:

$$\left| g(\boldsymbol{x}) - \sum_{i=1}^{k} \sum_{j=1}^{m_i} c_{ij} \sigma\left(w_{ij} y + \theta_{ij}\right) \right| < \varepsilon \tag{3.2.14}$$

■

Theorem 3.2.5 (Universality Theorem for $\sigma \in \mathcal{C}^\infty(\mathbb{R})$). *Shallow networks with an arbitrary number of units and activation function $\sigma \in \mathcal{C}^\infty(\mathbb{R})$ are universal in $\mathcal{C}(\mathbb{R}^n) \iff \sigma$ is not a polynomial.*

PROOF. (* only ($\Longleftarrow$) implication)
$\boxed{\Longleftarrow}$(*)

On behalf of lemma 3.2.4, we only need to do the proof for $n = 1$.

Consider w and b fixed. For any $h > 0$, we have that

$$\frac{\sigma\left(x(w+h)+b\right) - \sigma\left(xw+b\right)}{h} \in \mathcal{S}_1 \tag{3.2.15}$$

Hence, the limit for $h \to 0$ is in its closure $\overline{\mathcal{S}_1}$. This limit is $\frac{\partial}{\partial w}(\sigma\left(xw+b\right))$. We can apply an analogous argument using the same argument to prove that any k-th derivative is in $\overline{\mathcal{S}_1}$ (see fig. 3.1 for a visual explanation). Applying differentiation rules we have that:

$$\frac{\partial^k}{\partial w^k}(\sigma(xw+b)) = x^k \sigma^{(k}(xw+b) \in \overline{\mathcal{S}_1} \tag{3.2.16}$$

Since σ is not a polynomial, lemma 2.4.1 guarantees that there exists a number $\tilde{b}$ such that for any k, $\sigma^{(k}(\tilde{b}) \neq 0$. Taking $w = 0$ and $b = \tilde{b}$, we have that for any integer k, the monomial $x_k \in \overline{\Sigma_1}$. As a consequence, $\overline{\mathcal{S}_n}$ contains all polynomials.

By Stone-Weierstrass's theorem [21, Ch. 3], its closure contains all continuous functions.

$\boxed{\Longrightarrow}$

We prove by contradiction. Suppose σ is a polynomial of degree k. Then $\sigma(\langle \boldsymbol{x}, \boldsymbol{w}\rangle + b)$ is a polynomial degree at most k for any $\boldsymbol{w}$ and b. Thus, the family Σ_n is contained in the family of polynomials of degree at most k, which is not dense in $\mathcal{C}(\mathbb{R}^n)$. This is a classical density result, but we will give the details for the sake of completeness, for the case $n = 1$.

Suppose we have a sequence of polynomials of degree at most k: $\{P_i(\cdot)\}_{i\in\mathbb{N}}$, $P_i(x) = \sum_{j=1}^k a_j^i x^j$ such that

$$P_i(\cdot) \xrightarrow[i\to\infty]{} f$$

We consider two possibilities:

i $\forall j$, $\exists a^j \in \mathbb{R}$ such that $a_j^i \xrightarrow[i\to\infty]{} a_j$. In this case we will prove that $f(x) = a_0 + a_1 x + \cdots + a_k x^k$.

ii $\exists j_0$ such that $\{a_{j_0}^i\}_{i\in\mathbb{N}}$ has no limit. We will prove this case is in contradiction with $\{P_i\}_{i\in\mathbb{N}}$ being convergent.

In the first case, let us define $M = \max_{j=0,\dots,k} \|x^j\|$. Given $\varepsilon > 0$, we can choose i_0 such that

$$|a_j^i - a_j| < \frac{\varepsilon}{(k+1)M} \quad \forall i \geq i_0 \, \forall j = 0, \dots, k \tag{3.2.17}$$

If we define $g(x) = a_0 + a_1 x + \cdots + a_k x^k$, if follows $\forall i > i_0$:

$$\begin{aligned}
\|g - P_i\| &= \|(a_0 - a_0^i) + \cdots + (a_k - a_k^i)x^k\| && && (3.2.18)\\
&\leq |a_0 - a_0^i| + \cdots + |a_k - a_k^i|\|x^k\| && \text{Triangular inequality} && (3.2.19)\\
&\leq (|a_0 - a_0^i| + \cdots + |a_k - a_k^i|)M && \text{Definition of } M && (3.2.20)\\
&\leq \left(\frac{\varepsilon}{(k+1)M} + \cdots + \frac{\varepsilon}{(k+1)M}\right) M = \varepsilon && \text{eq. (3.2.17)} && (3.2.21)
\end{aligned}$$

In conclusion, $P_i \xrightarrow[i\to\infty]{} g$, since the limit is unique, $f = g$.

In case (ii), since $P_i \xrightarrow[i\to\infty]{} f$ implies $P_i(x) \xrightarrow[i\to\infty]{} f(x)$ almost for any x, we can choose $k+1$ different numbers $y_0, \dots, y_k$ such that the convergence is pointwise, i.e.:

$$\begin{cases} a_0^i + a_1^i y_0 + \cdots + a_k^i y_0^k & \xrightarrow[i\to\infty]{} \quad f(y_0) \\ \qquad\qquad \vdots & \qquad\qquad \vdots \\ a_0^i + a_1^i y_k + \cdots + a_k^i y_k^k & \xrightarrow[i\to\infty]{} \quad f(y_k) \end{cases} \tag{3.2.22}$$

Using properties of limits, we can make a linear combination of these limits to get:

$$a_0^i \left(\sum_{j=0}^k \gamma_j\right) + a_1^i \left(\sum_{j=0}^k \gamma_j y_j\right) + \dots a_k^i \left(\sum_{j=0}^k \gamma_j y_j^k\right) \xrightarrow[i\to\infty]{} \sum_{j=0}^k \gamma_j f(y_j) \tag{3.2.23}$$

where γ_j are coefficients of the linear combination. If we can choose these coefficients such that

$$\sum_{j=0}^k \gamma_j y_j^l = \begin{cases} 1 & \text{if } l = j_0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2.24}$$

eq. (3.2.23) becomes $a_{j_0}^i \xrightarrow[i\to\infty]{} \sum_{j=0}^k \gamma_j f(y_j)$ which is a contradiction.

It is indeed possible to choose $\{\gamma_j\}_{j=0:k}$ as in eq. (3.2.24), because they are the solution of the system of equations

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ y_0 & y_1 & \dots & y_k \\ \vdots & & & \vdots \\ y_0^k & y_1^k & \dots & y_k^k \end{pmatrix} \begin{pmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_k \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \leftarrow j_0\text{-th position} \tag{3.2.25}$$

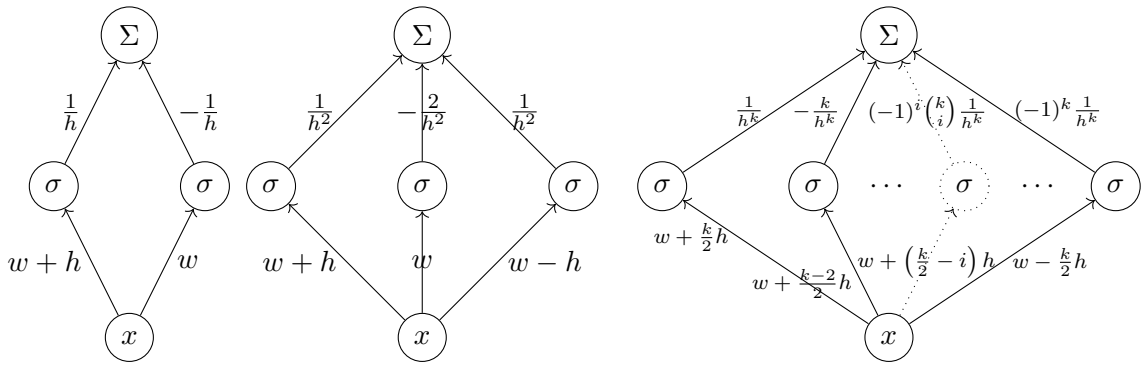


Figure 3.1: Illustration of how the k-th derivative is obtained using networks with k neurons (assuming $\sigma^{(k)}(0) \neq 0$). The input is represented by $\textcircled{x}$, each arrow represents a product and $\textcircled{\sigma}$ and $\textcircled{\Sigma}$ represent the activation and the sum functions respectively. So for example, the first picture represents $\frac{1}{h}\sigma(x(w+h)) - \frac{1}{h}\sigma(xw)$, which is the approximation for the first derivative $\frac{\partial}{\partial w}\sigma(xw) = x\sigma'(xw)$. We have explicitly added w, but in most cases we can suppose $w = 0$ (as explained in the proof of theorem 3.2.5.

The matrix of this system of equations is a Vandermonde matrix, therefore it has a unique solution if and only if all y_j are different.

We have proved that a sequence of polynomials of degree at most k can only have a polynomial of degree at most k as a limit. ■

With all these results, we are ready to extend the previous theorem to continuous activations, and therefore complete the proof of theorem 3.2.1.

PROOF. (Of theorem 3.2.1)
The left-to-right implication is analogous as in theorem 3.2.5. We will focus in the other implication and prove it by contradiction.

We define, for each $\varphi \in \mathcal{C}_0^\infty(\mathbb{R})$, the function $\sigma_\varphi = \sigma * \varphi$.

$$\sigma_\varphi(x) = \int_{-\infty}^{\infty} \sigma(x-y)\varphi(y)dy \tag{3.2.26}$$

Since $\sigma, \varphi \in \mathcal{C}(\mathbb{R})$ and φ has compact support, the integral converges for all x. Using lemma 2.3.1, $\sigma_\varphi \in \mathcal{C}^\infty(\mathbb{R})$. Taking Riemann sums, $\sigma_\varphi \in \overline{\mathcal{S}_n(\sigma)}$. Using this fact and that

$$\sigma_\varphi(wx+b) = \int_{-\infty}^{\infty} \sigma(wx+b-y)\varphi(y)dy \tag{3.2.27}$$

we have that $\overline{\mathcal{S}_n(\sigma_\varphi)} \subseteq \overline{\mathcal{S}_n(\sigma)}$. Because $\sigma_\varphi \in \mathcal{C}^\infty(\mathbb{R})$, we have from the proof of theorem 3.2.5, that

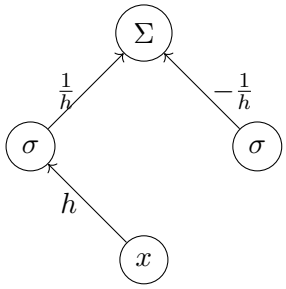


Figure 3.2: A special and interesting case happens when $b = 0$ and $\sigma'(0) \neq 0$. In this case, setting $w = 0$, the monomial x can be computed as $1/\sigma'(0)$ times the simple network of this figure. We consider this case specially interesting because it needs one less connection. This could be biologically more efficient. Moreover, this little network depends only on h, the parameter that gives the precision, so the same network could be used in many places of the brain where the basic monomial x is needed.

$x^k \sigma_\varphi^{(k}(b) \in \overline{\mathcal{S}_n(\sigma_\varphi)}$ for all $b \in \mathbb{R}$ and all $k \in \mathbb{N}$.

Suppose $\mathcal{S}_n(\sigma)$ is not dense in $\mathcal{C}(\mathbb{R})$, we will find a contradiction. In that case, there exists k such that $t^k \notin \overline{\mathcal{S}_n(\sigma)}$. Since for each $\varphi \in \mathcal{C}_0^\infty(\mathbb{R})$, $\overline{\mathcal{S}_n(\sigma_\varphi)} \subseteq \overline{\mathcal{S}_n(\sigma)}$, $t^k \notin \overline{\mathcal{S}_n(\sigma_\varphi)}$ for each φ. This implies that $\sigma_\varphi^{(k}(b) = 0$ for all $b \in \mathbb{R}$ and all $\varphi \in \mathcal{C}_0^\infty(\mathbb{R})$. Thus σ_φ is a polynomial of degree at most $k-1$ for each φ. If we set $\varphi_n = \eta_{1/n}$ as in lemma 2.3.2, the sequence $\{\sigma_{\varphi_n}\}_{n\in\mathbb{N}}$ tends to σ uniformly (by lemma 2.3.2) and by the proof of the left-to-right implication in theorem 3.2.5, a sequence of polynomials of degree at most $k-1$, if it converges, the limit is a polynomial of degree at most $k-1$. Thus σ is a polynomial, which is a contradiction. ■

Proposition 3.2.1 (Center of Mass Theorem). *Based on the idea appeared in the proof of theorem 3.2.5, of constructing polynomials from suitable derivatives, justify the following assertion regarding forgetting networks: Higher frequencies of the target function are forgotten faster than lower ones.*

PROOF.
Considering shallow networks with the forgetting hypothesis (section 1.2) to be that only weights (w's) are forgotten (i.e. $\varphi_a(t) = \varphi_b(t) = 1$ for all t) and a polynomial target function; then each monomial is forgotten as $x^k \to (\varphi(t) \cdot x)^k$, so high degree elements of the polynomial are forgotten much faster than small degree elements.

This fact is directly related with forgetting high frequencies faster than lower ones because from the perspective of polynomials, high frequencies are associated with high degree polynomials.

In deep networks the modeling of forgetting may be more complex since it can depend on the graph. Each layer may be forgotten in a different way, and even each neuron may have different $\varphi_a, \varphi_b, \varphi_w$. However, if we consider the forgetting hypothesis with $\varphi_b(t) = \varphi_w(t) = 1$ for all t, in the second layer we have a

phenomenon analogous to the forgetting hypothesis with $\varphi_a(t) = \varphi_b(t) = 1$ for all t in a single layer, so the same behavior of high frequencies directly applies.

These kind of arguments can be applied to gain intuition in how some specific networks forget, regarding high and low frequencies. ■

3.2.1 Calculate derivatives with neural networks

In this section we will give some intuition on the method to obtain polynomials using suitable derivatives, which is the basic idea in the proof of universality theorem.

First, we want to highlight the following related fact, that will also be useful in the proof of theorem 4.2.1.

Corollary 3.2.6. *Let H_k^1 be the set of single variable polynomials of degree most k. Then $H_k^1 \subseteq \overline{\mathcal{S}_{k+1,1}}$. And in general, for polynomials of n variables and degree k, $H_k^n \subseteq \overline{S_{s\left(\frac{k+n}{n}\right)^n,n}}$, where $s = \dim H_k^n = \binom{n+k-1}{k}$.*

PROOF.
In the proof of theorem 3.2.5 we have seen that $P_k^1 \subseteq \overline{\mathcal{S}_1}$ by saying that a multiple of the monomial x^k can be seen as a k-th derivative of $\sigma(wx + b)$ with respect to w for some $b \in \mathbb{R}$. This k-th derivative can be approximated by functions in $\mathcal{S}_1$. In particular, using the finite differences approach (for a visual representation see fig. 3.1, and for a developed theory, see [9, Ch. 3]) to approximate such derivative, a derivative of order k with can be approximated with $k+1$ evaluations of the function. In our context this means using $k + 1$ units.

For the multivariate case, we observe that in order to approximate the monomial $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ we need at most $\prod_{i=1}^n (\alpha_i + 1)$ units. This number follows from the subsequent reasoning: for each variable x_i, the monomial $x_i^{\alpha_i}$ can be approximated by a suitable network of $\alpha_i + 1$ units. If we do it for x_1, we can apply the finite differences method cited in the univariate case for the network obtained, so using $(\alpha_1 + 1) \cdot (\alpha_2 + 1)$ units the monomial $x_1^{\alpha_1} \cdot x_2^{\alpha_2}$ can be obtained. The same argument can be extended to n variables for any n.

Now we know $\sum_{i=1}^n (\alpha_i + 1) = k + n$, so using the inequality between arithmetic and geometric mean, we can find a bound for the number of units required for each monomial:

$$\left(\prod_{i=1}^n (\alpha_i + 1)\right)^{1/n} \leq \frac{k+n}{n} \implies \prod_{i=1}^n (\alpha_i + 1) \leq \left(\frac{k+n}{n}\right)^n \tag{3.2.28}$$

Since a base of H_k^n has exactly s monomials, that can be each approximated by networks with at most $\left(\frac{k+n}{n}\right)^n$ units, we have that $H_k^n \subseteq \overline{S_{s\left(\frac{k+n}{n}\right)^n,n}}$ as stated. ■

In this corollary the first of the bounds is optimum, because k-th derivatives cannot be approximated by finite differences with less than $k+1$ points. In contrast, the bound for the multivariate case is highly non optimal for two reasons. First one, the bound for $\prod_{i=1}^{n}(\alpha_i+1)$ is pretty strong, and in fact the equality happens only in a small number of cases, and only for k's that are multiple of n. The second reason is because we are not sure that our method for computing the multivariate derivative $\frac{\partial^k}{\partial^{\alpha_1}x_1\cdots\partial^{\alpha_n}x_n}$ uses the minimum possible number of neurons.

The method of the proof gives an easy rule to compute the number of neurons needed to approximate a given polynomial. As an example, the number of neurons needed to compute a simple polynomial as $xy+xy^2$ would be $2\cdot 2+2\cdot 3=10$.

Obviously, if one wants to generate all polynomials in n variables of degree at most k, the previous corollary can be applied adding the result homogeneous polynomials of degree for $i=0:k$. We think this number can be improved, and in fact, for the case of single variable polynomials, we have that $P_k^1 \subseteq \mathcal{S}_{2k+1,1}$. This is because the k-th derivative is obtained as the limit when $h\to 0$ of $\sum_{i=0}^{k}(-1)^i\binom{k}{i}\frac{1}{h^k}\sigma\left(x\left(w+h\left(\frac{k}{2}-i\right)\right)+b\right)$. The key observation is that, if k is even, the value of the parameters needed to compute the k-th derivative contains the values for all even numbers smaller than k, and if k is odd, contains all the values for odd numbers smaller than k. Then the linear combination of units can be properly arranged to compute any polynomial (see example below).

Example 4. We want to build a network that approximates $f(x)=x^3+3x^2+x+1$. Lemma 2.4.1 states that there exists $\tilde{b}$ such that $\sigma^{(k}(\tilde{b})\neq 0$. In this example, for the sake of simplicity, we suppose that $\tilde{b}=0$.

1. First we approximate each of the monomials:
   (a) $x^3 \approx \frac{1}{\sigma^{(3}(0)h^3}\Big(\sigma(3xh/2)-3\sigma(xh/2)+3\sigma(-xh/2)-\sigma(-3xh/2)\Big)$
   (b) $3x^2 \approx \frac{3}{\sigma^{(2}(0)h^2}\Big(\sigma(xh)+2\sigma(0)+\sigma(-xh)\Big)$
   (c) $2x \approx \frac{2}{\sigma'(0)h}\Big(\sigma(xh/2)-\sigma(-xh/2)\Big)$
   (d) $1 \approx \frac{1}{\sigma(0)}\Big(\sigma(0)\Big)$
2. Write the linear combination of the monomials

$$\begin{aligned} x^3+3x^2+x+1 \approx & \frac{1}{\sigma^{(3}(0)h^3}\Big(\sigma(3xh/2)-3\sigma(xh/2)+3\sigma(-xh/2)-\sigma(-3xh/2)\Big)+ \\ & +\frac{3}{\sigma^{(2}(0)h^2}\Big(\sigma(xh)+2\sigma(0)+\sigma(-xh)\Big)+ \\ & +\frac{2}{\sigma'(0)h}\Big(\sigma(xh/2)-\sigma(-xh/2)\Big)+\frac{1}{\sigma(0)}\Big(\sigma(0)\Big) \end{aligned}$$

and rearrange the terms

$$x^3+3x^2+x+1 \approx \frac{1}{\sigma^{(3}(0)h^3}\sigma(3xh/2)+\frac{3}{\sigma^{(2}(0)h^2}\sigma(xh)+\left(-3\frac{1}{\sigma^{(3}(0)h^3}+\frac{2}{\sigma'(0)h}\right)\sigma(xh/2)+$$
$$+\left(2\frac{3}{\sigma^{(2}(0)h^2}+\frac{1}{\sigma(0)}\right)\sigma(0)+\left(3\frac{1}{\sigma^{(3}(0)h^3}-\frac{2}{\sigma'(0)h}\right)\sigma(-xh/2)+$$
$$+\frac{-3}{\sigma^{(2}(0)h^2}\sigma(-xh)+\frac{-1}{\sigma^{(3}(0)h^3}\sigma(-3xh/2)$$

The following figure provides a graphical representation

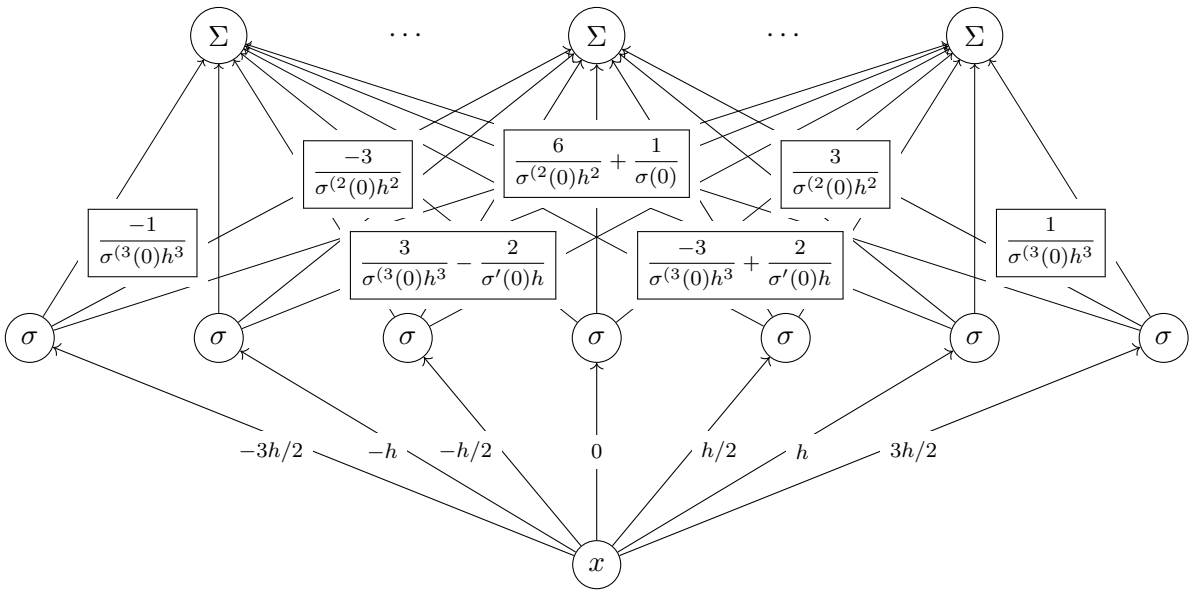


Figure 3.3: Approximation of $f(x) = x^3 + 3x^2 + x + 1$ using 7 units instead of 10. The other Σ's represent any other 3-order polynomial that can be generated with some combination of the same set of 7 neurons. As long as σ is such that $\sigma^{(k}(0) \neq 0$ for every k the above 7 "basis" can generate any 3-order polynomial. In general, with $2N + 1$ neurons we can generate any polynomial of order at most N.

3.3 Universality of Deep Networks

To directly apply theorem 3.2.1 to deep networks, we need the constituent functions in each node of the graph to be continuous [1]. This is what we define as internally continuous $\mathcal{G}$-functions:

[1] Or fulfill the more general condition of belonging to the set M defined in [24, Sec. 4].

Definition 3.3.1 (Internally continuous $\mathcal{G}$-function). A $\mathcal{G}$-function is said to be ***internally continuous*** if it admits some representation in which all its constituent functions are continuous. By analogy, we define ***internally*** $\mathcal{C}^k$ $\mathcal{G}$-functions.

A simple consequence of the universality theorem for shallow networks is the following:

Corollary 3.3.1 (Density of deep networks in internally continuous $\mathcal{G}$-functions). *Let $\mathcal{G}$ be a CDAG. Then $\mathcal{G}$-Deep networks with an arbitrary number of units and (possibly different in each node) activation functions $\{\sigma_v\}_{v\in\mathcal{V}} \subseteq \mathcal{C}^\infty$ are universal in $\mathcal{C}(\mathbb{R}^n) \cap \{$internally continuous $\mathcal{G}$-functions$\}$ $\iff$ for all $v \in \mathcal{V}$ σ_v is not a polynomial .*

The proof is simply done by applying the theorem to each constituent function. We omit more explicit details.

We want to note that in real applications, for this result to apply you need to guess the compositional form of the function you want to approximate beforehand. An interesting result would be to know whether deep networks are universal irrespective of their internal structure.

Corollary 3.3.1 can be refined considering that the required structure of the target function has to be only a subgraph of the network. So the result would be the same as in corollary 3.3.1, but the set in which networks with a given $\mathcal{G}$ structure are universal becomes:

$$\mathcal{C}(\mathbb{R}^n) \cap \{\text{internally continuous } \mathcal{H}\text{-functions, where } \mathcal{H} \text{ is a subgraph of } \mathcal{G}\}$$

We would have a *true* universality theorem for deep networks if, given a graph $\mathcal{G}$, all continuous functions had a decomposition that made them internally continuous $\mathcal{G}$-functions. There are reasons to think this is possible.

If no conditions are imposed to the constituent functions, all functions can be regarded as $\mathcal{G}$ functions, for any CDAG $\mathcal{G}$ with the right number of source nodes. This result will be formally stated later (proposition 3.3.2).

The proof of this statement, relies on bijective functions between $\mathbb{R}$ and $\mathbb{R}^n$ and its inverses. It is a well known topological fact that those functions cannot be continuous.

The question that naturally arises is whether constituent functions can be limited to be continuous. This is not a new problem, but as far as we know there is no answer to that. An important related result is ***Kolmogorov-Arnold representation theorem*** ([4, 18]), which solved Hilbert's 13^{th} problem and states that any continuous multivariate function $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ has a decomposition of the form:

$$f(x_1, \ldots, x_n) = \sum_{q=1}^{2n} \Phi_q \left(\sum_{p=1}^{n} \varphi_{q,p}(x_p) \right) \qquad \Phi_q, \varphi_{q,p}\text{continuous} \tag{3.3.1}$$

This result requiring only one function Φ would directly state that any continuous function is in fact an internally continuous function. Although we have found no answer to the proposed question, we have found a paper from Giorsi and Poggio [13] stating that Kolmogorov's theorem is irrelevant because constituent functions are continuous but highly non-smooth, while there is another paper by Kůrková [19] satating that Kolmogorov theorem is indeed relevant.

As commented previously, if no condition is imposed to the constituent functions, then the following result holds:

Proposition 3.3.2. *For any CDAG $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and any function $f : \Omega \to \mathbb{R}$, there exists a set of constituent functions $\{h_v\}_{v\in\mathcal{V}}$ such that f is a $\mathcal{G}$ function with $\{h_v\}_{v\in\mathcal{V}}$ as constituent functions.*

This is an intuitive observation if one keeps in mind the existence of bijective functions between $\mathbb{R}$ and $\mathbb{R}^n$. We will not give explicit details on this result.

If more-that-continuous regularity is imposed, proposition 3.3.2 does not hold. In fact, there is a more general theory on this aspect developed by Vitushkin (see [13, Th. 2.1]) who states the following theorem:

Theorem 3.3.3. *For any pair of natural numbers $k \geq 1$ and $n \geq 2$, there exist functions $f \in \mathcal{C}^k(I^n)$ that cannot be expressed as a superposition and composition of functions C^k of $n-1$ variables.*

The proof of this result is very much beyond the scope of this thesis, but we present the proof of a weaker versions:

Proposition 3.3.1. *Prove that for $\mathcal{G}$ the binary tree of 4 nodes, there exist continuous functions $f \in \mathcal{C}(\mathbb{R}^4)$ such that f are not internally $\mathcal{C}^\infty$ $\mathcal{G}$-functions.*

PROOF.
With the above mentioned graph, a $\mathcal{G}$-function f has a decomposition of the form

$$f(x_1, x_2, x_3, x_4) = h\big(g_1(x_1, x_2), g_2(x_3, x_4)\big) \tag{3.3.2}$$

being its constituent functions $h, g_1, g_2 \in \mathcal{C}^2(\mathbb{R}^4)$. If we differentiate f with respect to x_1

$$\partial_{x_1} f(\boldsymbol{x}) = \partial_{y_1} h(g_1(x_1, x_2), g_2(x_3, x_4)) \cdot \partial_{x_1} g_1(x_1, x_2) \tag{3.3.3}$$

And differentiating again, this time with respect to x_3

$$\partial_{x_1 x_3} f(\boldsymbol{x}) = \partial_{y_1 y_2} h(g_1(x_1, x_2), g_2(x_3, x_4)) \cdot \partial_{x_3} g_2(x_3, x_4) \cdot \partial_{x_1} g_1(x_1, x_2) \tag{3.3.4}$$

By analogy, $\partial_{x_1 x_4} f(\boldsymbol{x})$ is

$$\partial_{x_1 x_4} f(\boldsymbol{x}) = \partial_{y_1 y_2} h(g_1(x_1, x_2), g_2(x_3, x_4)) \cdot \partial_{x_4} g_2(x_3, x_4) \cdot \partial_{x_1} g_1(x_1, x_2) \tag{3.3.5}$$

Now considering the quotient between (3.3.4) and (3.3.5)

$$\frac{\partial_{x_1 x_3} f(\boldsymbol{x})}{\partial_{x_1 x_4} f(\boldsymbol{x})} = \frac{\partial_{x_3} g_2(x_3, x_4)}{\partial_{x_4} g_2(x_3, x_4)} \tag{3.3.6}$$

Since the RHS does not depend on x_1, the LHS cannot depend on x_1. Therefore $\partial_{x_1}\left(\frac{\partial_{x_1 x_3} f(\boldsymbol{x})}{\partial_{x_1 x_4} f(\boldsymbol{x})}\right) = 0$. Using derivation formulas, the numerator of $\partial_{x_1}\left(\frac{\partial_{x_1 x_3} f(\boldsymbol{x})}{\partial_{x_1 x_4} f(\boldsymbol{x})}\right)$ is

$$\left(\partial_{x_1 x_3 x_1} f(\boldsymbol{x})\right) \cdot \left(\partial_{x_1 x_4} f(\boldsymbol{x})\right) - \left(\partial_{x_1 x_4 x_1} f(\boldsymbol{x})\right) \cdot \left(\partial_{x_1 x_3} f(\boldsymbol{x})\right) = 0 \tag{3.3.7}$$

Now, it gets easy to find a function that does not satisfy that condition. For example, set $f(x_1, x_2, x_3, x_4) = x_1 x_3 + x_1^2 x_4$, it clearly does not fulfill the given condition. ■

Chapter 4

Curse of Dimensionality

4.1 Framework and Definitions

In the previous chapter we have studied universality results. The problem of these results is that they are true when there is no limit to the number of units in the network. In a real world situation, the number of units is constrained by computational limitations.

In this chapter we will study how the number of neurons needed varies with the accuracy imposed to the network and the dimension of the problem, measured as the dimension of the input (n in definition 1.1.2).

In particular, we will compare the case of shallow and deep networks. We will show that for a fixed accuracy, the number of neurons needed to ensure a shallow network can achieve that given accuracy grows exponentially with n, whereas in deep networks the growth is linear, making the second case computationally much more efficient, assuming some structure of the target functions.

We will formulate the results in terms of *best approximation error*, defined in definition 2.4.1.

With these terms, the problem we will focus in the chapter can be formulated as: given $\varepsilon > 0$, a subset of target functions $W \subseteq \mathbb{X}$, and $\mathcal{S}_{N,n} \subseteq \mathbb{X}$ the set of neural networks with less or equal than N units, which is the minimum N such that

$$E\left(W; \mathcal{S}_{N,n}\right) < \varepsilon \tag{4.1.1}$$

We will use asymptotic notation in this section. For example, if $\operatorname{dist}(f, \mathcal{S}_{N,n}) = \mathcal{O}\left(N^{-\gamma}\right)$ for some $\gamma > 0$, then a network with complexity $N = \mathcal{O}\left(\varepsilon^{-\frac{1}{\gamma}}\right)$ is enough to guarantee an approximation of accuracy at least ε.

The results we present are based on [30] and [28]. They rely on the fact that the space W of target functions is somehow controlled. We will think of Sobolev spaces as in definition 2.2.3 with the sup norm

and over $\Omega = [-1,1]^n$, so we will generally write W_m^n instead of $W_{\infty,m}^n([-1,1]^n)$.

4.2 Best Approximation for Shallow Networks. Curse of Dimensionality

Theorem 4.2.1 (Curse of dimensionality). *Let $(\mathbb{X}, \|\cdot\|_p)$ be a normed space with $p \in [1,\infty]$. Let $\sigma \in \mathcal{C}^\infty(\mathbb{R})$ and not a polynomial. For $f \in W_{\infty,m}^n([-1,1]^n)$, the complexity of the shallow networks that provide accuracy at least ε is*

$$N = \mathcal{O}\left(\varepsilon^{-n/m}\right) \tag{4.2.1}$$

and it is the best possible among all reasonable methods of approximation. By reasonable we mean the ones described after the definition of non-linear N-width.

PROOF.
The first proof of this result was given in [27]. We will proceed with a slightly different approach, following [28], we will prove that $E(\mathrm{W}_m^n; \mathcal{S}_{N,n}) \leq CN^{-m/n}$ for a suitable constant C independent of N.

We begin recalling corollary 3.2.3 and corollary 3.2.6, that together tell us

$$P_k^n = \left\{ \sum_{i=1}^{s} g_i(\mathbf{a}^i \cdot \mathbf{x}) \ : \ g_i \in \overline{\mathcal{S}_{k+1,n}} \ \forall i \right\} \tag{4.2.2}$$

where $s = \dim P_k^n = \binom{n+k}{k}$.

From this result it follows directly that:

$$P_k^n \subseteq \overline{\mathcal{S}_{s(k+1),n}} \tag{4.2.3}$$

Set $N = s(k+1)$. Then there exists a constant C' independent of N such that

$$\begin{aligned} E(\mathrm{W}_m^n; \mathcal{S}_{N,n}) &= E(\mathrm{W}_m^n; \overline{\mathcal{S}_{N,n}}) && && (4.2.4)\\ &\leq E(\mathrm{W}_m^n; P_k^n) && \text{eq. (4.2.3)} && (4.2.5)\\ &\leq C' k^{-m} && \text{lemma 2.4.2} && (4.2.6) \end{aligned}$$

For n fixed and k growing ($k \gg n$), which corresponds to greater accuracy, we have that $N = \Theta(k^n)$, so from the last equation we can say there exists a constant C independent of N such that

$$E(\mathrm{W}_m^n; \mathcal{S}_{N,n}) \leq C' k^{-m} \leq CN^{-m/n} \tag{4.2.7}$$

We still have to prove that this is the (asymptotically) best possible complexity. As we have explained before, this is equivalent to prove that:

$$h_{(n+2)N}(W_m^n; c) \geq CN^{-m/n} \tag{4.2.8}$$

for some constant C. This is a direct consequence of lemma 2.4.3 and lemma 2.4.4. The application of both lemmas gives:

$$h_{(n+2)N}(W_m^n; c) \geq C\left(N(n+2)\right)^{-m/n} = CN^{-m/n}(n+2)^{-n/m} \tag{4.2.9}$$

This extra factor $(n+2)^{-n/m}$ is asymptotically negligible because $\lim_{n\to\infty}(n+2)^{-n/m} = 1$. ■

A useful and direct consequence of this theorem is the corresponding result for polynomials.

Corollary 4.2.2 (Polynomials version). *With the same hypothesis as theorem 4.2.1 but restricting $f \in P_k^n$, any f can be approximated with arbitrary accuracy by shallow networks with exactly $N = (k+1)\binom{k+n-1}{k} \approx k^n$ units.*

PROOF.
This is a direct consequence of eq. (4.2.3). ■

4.3 Deep Networks Avoid the Curse of Dimensionality

As was said before, for deep networks we will need to assume a certain structure of the graph corresponding to the target function. In particular, we will assume functions to have a regular tree structure.

Definition 4.3.1 (K-ary tree, $W_m^{n,K}$ space). A CDAG is said to be a K-ary tree if each node has no more than K in-edges. We define $W_m^{n,K}$ to be a subset of W_m^n such that:

$$W_m^{n,K} \stackrel{\text{def}}{=} \left\{ f \in \mathrm{W}_m^n \ : \ \begin{array}{c} f \text{ is a } \mathcal{G}\text{-function with } \mathcal{G} \text{ a } K\text{-ary tree} \\ \text{and constituent functions } h \in \mathrm{W}_m^K \end{array} \right\} \tag{4.3.1}$$

Note that a K-ary tree is a layered graph (definition 1.1.5).

The analogous theorem for deep networks is presented and proved in [30] for the case of a binary tree ($K = 2$). We state it in a more general form, but the ideas behind are essentially the same.

Theorem 4.3.1. *Let $\sigma \in \mathcal{C}^\infty(\mathbb{R})$ and not a polynomial. For $f \in \mathrm{W}_m^{n,K}$, the complexity of the deep networks that provide accuracy at least ε with the **sup** norm is*

$$N = \mathcal{O}\left((n-1)\,\varepsilon^{-K/m}\right) \tag{4.3.2}$$

PROOF.
We prove this theorem by induction on d the number of hidden layers of the associated graph. The base case is equivalent to theorem 4.2.1.

Consider it to be true for networks of less than d layers (induction hypothesis), we will prove the theorem for networks of exactly d layers. By theorem 4.2.1 each of the constituent functions of f can be approximated up to accuracy ε with $\mathcal{O}\left(\varepsilon^{-K/m}\right)$ units.

We wish to remark that the constituent functions of the network are Lipschitz continuous (since they are continuously differentiable in the compact set Ω). In fact, due to the norm restriction of derivatives in W_m^n and to mean value theorem, this Lipschitz constant is at most 1. We will use this fact in the proof.

If we take h to be the constituent function of the sink node, $h_1, \ldots, h_K$ the constituent functions of the layer below (the last hidden layer) and $P, P_1, \ldots, P_K$ the shallow networks approximating those mentioned constituent functions with accuracies

$$\|h - P\| \leq \frac{\varepsilon}{2} \qquad \|h_i - P_i\| \leq \frac{\varepsilon}{2K} \tag{4.3.3}$$

Then using Minkowskii inequality we have:

$$\|h(h_1, \ldots, h_K) - P(P_1, \ldots, P_K)\| = \|h(h_1, \ldots, h_K) - h(P_1, \ldots, P_K) + h(P_1, \ldots, P_K) - P(P_1, \ldots, P_K)\| \tag{4.3.4}$$

$$\leq \|h(h_1, \ldots, h_K) - h(P_1, \ldots, P_K)\| + \|h(P_1, \ldots, P_K) - P(P_1, \ldots, P_K)\| \tag{4.3.5}$$

The second summand, by eq. (4.3.3) is less or equal than $\frac{\varepsilon}{2}$. [1] The first one is bounded as follows:

$$\|h(h_1, \ldots, h_K) - h(P_1, \ldots, P_K)\| \leq \|(h_1 - P_1, \ldots, h_K - P_K)\| \qquad \text{(Lipschitz)} \tag{4.3.6}$$

$$\leq \sum_{i=1}^{K} \|h_i - P_i\| \qquad \text{(Triangular inequality)} \tag{4.3.7}$$

$$\leq K \cdot \frac{\varepsilon}{2K} = \frac{\varepsilon}{2} \qquad \text{(eq. (4.3.3))} \tag{4.3.8}$$

From eq. (4.3.4) it directly follows that

$$\|h(h_1, \ldots, h_K) - P(P_1, \ldots, P_K)\| \leq \varepsilon \tag{4.3.9}$$

as desired. Now by theorem 4.2.1, the first approximation in eq. (4.3.3) can be obtained with $\mathcal{O}\left(\left(\frac{\varepsilon}{2}\right)^{-K/m}\right) = \mathcal{O}\left(\varepsilon^{-K/m}\right)$ units. Since the h_i's can be considered as deep networks of $d-1$ hidden layers, each of the approximations to h_i, by induction can be obtained with $\mathcal{O}\left(\left(\frac{n}{K} - 1\right)\left(\frac{\varepsilon}{2K}\right)^{-K/m}\right) = \mathcal{O}\left(\left(\frac{n}{K} - 1\right)\varepsilon^{-K/m}\right)$.

Because there are K nodes in the final layer, the total number of units needed is indeed $\mathcal{O}\left(\varepsilon^{-K/m}\right) + K\mathcal{O}\left(\left(\frac{n}{K} - 1\right)\varepsilon^{-K/m}\right) = \mathcal{O}\left((n-1)\varepsilon^{-K/m}\right)$ ■

[1] It is interesting to note that here the ***sup*** norm is important. This statement will not be true in general for another $\mathcal{L}^p$ norm.

4.4 Comments and Generalizations

Both theorem 4.2.1 and theorem 4.3.1 are only valid for $\mathcal{C}^\infty$ activation functions. The ReLU function (section 1.1), which is one of the most used as an activation function, does not fall into this category. We don't believe this is a serious limitation because one can find arbitrarily close functions to it.

For the case of shallow networks, very similar results have been proved for continuous (but not differentiable) activation functions considering the $\mathcal{L}^2$ norm, but they cannot be extended to deep networks using the techniques of theorem 4.3.1 because the proof of theorem 4.3.1 is only valid for the sup norm. We will not comment these result, the interested reader is referred to [30, Section 4].

Proposition 4.4.1. *Consider a forgetting shallow network $\Sigma(\boldsymbol{x}; t)$ with all forgetting functions $\varphi_a, \varphi_b, \varphi_w$ equal and note them as $\varphi(t)$. Consider also that the activation function σ is the ReLU. Show that*

$$\Sigma(\boldsymbol{x}; t) = \varphi^2(t)\Sigma(\boldsymbol{x}; t = 0) \tag{4.4.1}$$

PROOF.
Since it is the most commonly used we will consider the activation function σ to be the ReLU function, this is, $\sigma(x) = \max(0, x)$. Since this function is linear with respect to positive number (this is called ***positive homogeneity***, i.e. if a is a positive number, $\sigma(ab) = a\sigma(b)$) and $\varphi(t)$ is always a positive number, equation (1.2.1) directly becomes:

$$\Sigma(\boldsymbol{x}; t) = \sum_{k=1}^{N} a_k \varphi(t) \sigma(\langle \boldsymbol{x}, \boldsymbol{w}_k \varphi(t) \rangle + b_k \varphi(t)) \tag{4.4.2}$$

$$= \sum_{k=1}^{N} a_k \varphi^2(t) \sigma(\langle \boldsymbol{x}, \boldsymbol{w}_k \rangle + b_k) \tag{4.4.3}$$

$$= \varphi^2(t)\Sigma(\boldsymbol{x}; t = 0) \tag{4.4.4}$$

■

We want to generalize this result to deep networks. To do so we propose a reasonable and simple biological mechanism for this behavior. To explain it, let us discuss the role of biases in the network. Biases can be seen as thresholds. In the ReLU case, since a neuron is activated when the logit is positive, we may assume that biases are generally negative, otherwise a neuron with null input would be activated. Therefore, the effect of multiplying by a number between 0 and 1 (only possible values of $\varphi(t)$) will actually ***increase*** the value of the bias, or equivalently, setting a lower threshold. Biologically speaking, when one layer of the network forgets with some rate, the following layers detect a lower signal than before, and so they tune their biases in a similar way the signal is lowered, to compensate the effect.

Proposition 4.4.2. *With these hypothesis, show that, for a forgetting deep network of d layers*

$$\Sigma(\mathbf{x}; t) = \left(\prod_{k=1}^{d} \varphi_k^2(t) \right) \Sigma(\mathbf{x}; t = 0) \tag{4.4.5}$$

Also show that, for the specific case of $\varphi_k(t) = e^{-t/\tau_k}$ being exponential decay with a different time constant for each layer τ_k, then the whole layer forgets exponentially, with an equivalent time constant of

$$\tau_{eq} = \frac{1}{\sum_{i=1}^{d} \frac{2}{\tau_i}} \tag{4.4.6}$$

PROOF.
Suppose we have a network of d layers, and the i-th layer forgets with a forgetting function $\varphi(t)$. As we have shown before in eq. (4.4.4), the output of that layer will be multiplied by $\varphi^2(t)$. This is the input of the following layer, so in the $(i+1)$-th layer, the input will be multiplied by $\varphi^2(t)$. The bias will also be multiplied by $\varphi^2(t)$ by hypothesis. Using the symmetric property of the scalar product and an analogous reasoning as in eq. (4.4.4), the output of the $(i+1)$-th layer is multiplied by $\varphi^2(t)$. Repeating this reasoning with the layers above, we reach the conclusion that this $\varphi^2(t)$ affects directly to the result of the whole network.

One can observe that this behavior is additive, meaning that if another layer, say the j-th one, forgets at a given rate $\varphi'(t)$ that could be different from $\varphi(t)$, and all biases of layers $(j+1), \ldots, d$ are tuned by the way described before, then the output is multiplied by and extra $\varphi'^2(t)$.

In general, if we have a different forgetting function for each layer, $\varphi_k(t)$ for $k = 1, \ldots, d$, then the whole network Σ behaves as stated in eq. (4.4.5), as we wanted to show.

If we substitute now $\varphi_k(t)$ for each decreasing exponential, the product factor in eq. (4.4.5) becomes a single exponential with the sum of corresponding exponents. This leads directly to eq. (4.4.6).

We want to note that

$$\tau_{eq} = \frac{1}{\sum_{i=1}^{d} \frac{2}{\tau_i}} = \frac{1}{2d} \frac{d}{\sum_{i=1}^{d} \frac{1}{\tau_i}} \tag{4.4.7}$$

is $\frac{1}{2d}$ times the harmonic mean of $\tau_1, \ldots, \tau_d$. This mean gives a heavier weight to small values than the common arithmetic mean, implying that the layers that forget faster (that is, with smaller τ) are the ones to determine the final forgetting behavior of the network. ■

Part II

Supervised Learning Algorithms

Chapter 5

Non Existence of Universal Learning Algorithm: No Free Lunch Theorems

5.1 Brief History and Justification

In this chapter we will present some of the so-called ***No Free Lunch Theorems*** (NFL). This theorems were developed by Wolpert and Macready in the 90's [36, 39, 40]. They give some general results that can be synthesized as: "there is no universal optimization algorithm ".

We will focus in two of the results known as NFL theorems: one general result for optimization theory, and a second one specific for learning algorithms. We find the first one relevant in the context of machine learning because in the core of any ML algorithm, there is a loss function to be minimized.

5.2 NFL for General Optimization

5.2.1 Framework

Let $\mathcal{X}$ be the search space, this is the space of possible outputs of the algorithm. Typically it can be considered that $\mathcal{X} = \mathbb{R}^n$. Let $\mathcal{Y}$ be the cost space, this is the space of all possible cost values. Typically it can be considered $\mathcal{Y} = \mathbb{R}$. An optimization problem is a function $f : \mathcal{X} \to \mathcal{Y}$. It is a common notation to define the set of all problems as $\mathcal{Y}^{\mathcal{X}} \stackrel{\text{def}}{=} \{f : \mathcal{X} \to \mathcal{Y}\}$.

We will call a time ordered set of m distinct points of $\mathcal{X}$ a sample of size m, and we will denote them as

$$d_m \stackrel{\text{def}}{=} \{(d_m^x(1), d_m^y(1)), \ldots, (d_m^x(m), d_m^y(m))\} \quad d_m^x(i) \in \mathcal{X}, d_m^y(i) \in \mathcal{Y} \tag{5.2.1}$$

In a sample $d_m^x(i)$ indicates the point generated in the i-th iteration, and $d_m^y(i)$ indicates its corresponding cost. With these notation, we define the following sets:

$$\mathcal{D}_m \stackrel{\text{def}}{=} (\mathcal{X}, \mathcal{Y})^m, \qquad \mathcal{D} \stackrel{\text{def}}{=} \cup_{m \geq 0} D_m \tag{5.2.2}$$

With this notation, and optimization algorithm can be defined as a function $a : \mathcal{D} \to \mathcal{X}$. Note that the $(m+1)$-th point can depend on all m previous points. We will demand our algorithms to avoid repeating points. Formally, if $a(d_m) = \hat{x}$, then for all i, $d_m^x(i) \neq \hat{x}$. The performance of an algorithm after m iterations will be given by $d_m^y(m)$. Depending on the problem, the measure will be different. For example, if it is a minimization problem, the smaller the value of $d_m^y(m)$ the better. We will not care of performance measure, we will only need the performance to depend only on $d_m^y(m)$.

For simplicity of the argument, as in [40], we will consider a discretized version of the problem. This means, we will consider $\mathcal{X}$ and $\mathcal{Y}$ to be ***finite spaces***. This is realistic assumption in the context of computer algorithms, since all quantities will be encoded with a finite number of bits, so in the end of the day we are not working with $\mathbb{R}$ or $\mathbb{R}^n$ but with a finite discretization of them.

Also as in [40] we will consider a probabilistic approach. We will consider, given an optimization problem f, a number of steps m, an algorithm a and a sample d_m the conditional probability of an obtaining sample costs d_m^y:

$$\Pr(d_m^y | f, m, a) \tag{5.2.3}$$

This means eq. (5.2.3) is the probability of obtaining sample costs d_m^y when initialization is chosen at random, provided the optimization problem is f, the algorithm is a and the number of steps is m.

Since we are taking this finite approach, this probabilities will be in general different from zero, and it will make sense to consider the sums. If we had a continuous approach, these probabilities will be zero a.e. and we would need to consider integrals of the corresponding density function to obtain relevant results.

5.2.2 Statement and Proof

We are now prepared to establish the first NFL result:

Theorem 5.2.1 (No Free Lunch, Th. 1 in [40]). *For any sample costs d_m^y of size m, and any algorithms a_1, a_2*

$$\sum_{f \in \mathcal{Y}^{\mathcal{X}}} \Pr\left(d_m^y | f, m, a_1\right) = \sum_{f \in \mathcal{Y}^{\mathcal{X}}} \Pr\left(d_m^y | f, m, a_2\right) \tag{5.2.4}$$

PROOF.
The proof is done by induction on the number of steps m. For $m = 1$, the sample is $d_1 = (d_1^x, d_1^y)$. The only possible value for the cost is $f(d_1^x)$. In this case

$$\sum_{f \in \mathcal{Y}^{\mathcal{X}}} \Pr(d_1^y | f, m = 1, a) = \sum_{f \in \mathcal{Y}^{\mathcal{X}}} \delta(d_1^y, f(d_1^x)) \tag{5.2.5}$$

where δ represents the Kronecker delta function [1]. Equation (5.2.5) can be computed using finiteness of $\mathcal{X}$ and $\mathcal{Y}$ as the number of functions $f : \mathcal{X} \to \mathcal{Y}$ such that $f(d_1^x) = d_1^y$, fixed (d_1^x, d_1^y). This number is $|\mathcal{Y}|^{|\mathcal{X}|-1}$ independent of a.

Now let's prove the induction step. To do so, consider a sample d_{m+1} of size $m+1$. We will split its corresponding probability, using the definition of conditional probability, in the following way:

$$\Pr\left(d_{m+1}^y | f, m+1, a\right) = \Pr\left(d_{m+1}^y(m+1) | d_m, f, m+1, a\right) \cdot \Pr(d_m^y | f, m, a) \tag{5.2.6}$$

where $d_{m+1}^y = (d_{m+1}^y(1), \ldots, d_{m+1}^y(m+1))$ and $d_m^y = (d_{m+1}^y(1), \ldots, d_{m+1}^y(m))$.

Now consider $x \in \mathcal{X}$ the new value of the search space. We will expand the corresponding probability in the left factor of eq. (5.2.6) over all possible values of x:

$$\begin{aligned}
\Pr\left(d_{m+1}^y(m+1) | d_m, f, m+1, a\right) &= \sum_{x \in \mathcal{X}} \Pr(d_{m+1}^y(m+1) | f, x) \cdot \Pr(x | d_m, f, m+1, a) && (5.2.7)\\
&= \sum_{x \in \mathcal{X}} \delta\big(d_{m+1}^y(m+1), f(x)\big) \cdot \Pr(x | d_m, f, m+1, a) && (5.2.8)\\
&= \sum_{x \in \mathcal{X}} \delta\big(d_{m+1}^y(m+1), f(x)\big) \cdot \delta(x, a(d_m)) && (5.2.9)\\
&= \delta\big(d_{m+1}^y(m+1), f(a(d_m))\big) && (5.2.10)
\end{aligned}$$

Now considering eq. (5.2.6) and eq. (5.2.10), the sum we are interested in has the following form:

$$\sum_{f \in \mathcal{Y}^{\mathcal{X}}} \Pr\left(d_{m+1}^y | f, m+1, a\right) = \sum_{f \in \mathcal{Y}^{\mathcal{X}}} \delta\big(d_{m+1}^y(m+1), f(a(d_m))\big) \cdot \Pr(d_m^y | f, m, a) \tag{5.2.11}$$

Since we are considering that the algorithm does not repeat points (and therefore $a(d_m) \notin d_m^x$) [2], the left factor depends only on the value of f for points outside d_m^x, while the right factor depends only on values of f for points in d_m^x. So if we consider the auxiliary sets

$$\mathcal{Y}^{\mathcal{X} \setminus d_m^x} \stackrel{\text{def}}{=} \{f : \mathcal{X} \setminus d_m^x \to \mathcal{Y}\} \qquad \mathcal{Y}^{d_m^x} \stackrel{\text{def}}{=} \{f : d_m^x \subseteq \mathcal{X} \to \mathcal{Y}\} \tag{5.2.12}$$

[1] Defined as $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ if $x \neq y$.

[2] Here we are using the abuse of notation $d_m^x \stackrel{\text{def}}{=} \{d_m^x(1), \ldots, d_m^x(m)\} \stackrel{\text{def}}{=} \{d_{m+1}^x(1), \ldots, d_{m+1}^x(m)\}$.

we can split sum in eq. (5.2.11) in the following way: [3]

$$\sum_{f\in\mathcal{Y}^{\mathcal{X}}} \Pr\left(d^y_{m+1}|f,m+1,a\right) = \left(\sum_{f\in\mathcal{Y}^{\mathcal{X}\setminus d^x_m}} \delta(d^y_{m+1}(m+1), f(a(d_m)))\right)\cdot\left(\sum_{f\in\mathcal{Y}^{d^x_m}} \Pr(d^y_m|f,m,a)\right) \quad (5.2.13)$$

By an analogous argument as in the base case, the left factor equals $|\mathcal{Y}|^{|\mathcal{X}|-m-1}$. To calculate the value of the second factor, consider the sum $\sum_{f\in\mathcal{Y}^{\mathcal{X}}} \Pr(d^y_m|f,m,a)$ that does not depend on algorithm a because of the induction hypothesis. Since the summands depend only on the values of f for points in d^x_m, it can be written as:

$$\sum_{f\in\mathcal{Y}^{\mathcal{X}}} \Pr(d^y_m|f,m,a) = |\mathcal{Y}|^{|\mathcal{X}|-m}\left(\sum_{f\in\mathcal{Y}^{d^x_m}} \Pr(d^y_m|f,m,a)\right) \quad (5.2.14)$$

So in the end we can write

$$\sum_{f\in\mathcal{Y}^{\mathcal{X}}} \Pr\left(d^y_{m+1}|f,m+1,a\right) = |\mathcal{Y}|^{-1}\left(\sum_{f\in\mathcal{Y}^{\mathcal{X}}} \Pr(d^y_m|f,m,a)\right) \quad (5.2.15)$$

and by the induction hypothesis, does not depend on a. ■

The analogous result considering cost functions that vary with time is also stated and proved in [40, Th.2].

5.2.3 Discussion

There are some points to remark concerning the relevance of this result in the context of learning algorithms:

(a) It supposes that the algorithm never reaches a fixed point. In an optimization framework, where there is a point with minimum cost, this means that for an algorithm to fulfill theorem 5.2.1's hypothesis, it has to escape from the minimum. Since the cost is computed using the data of all visited points, escaping will not reduce the overall cost, but makes the algorithm less intuitive.

[3] In a general setting, suppose we have sets $\mathcal{F},\mathcal{A},\mathcal{B}$ that are related by a bijection σ: $\mathcal{F}\overset{\sigma}{\longleftrightarrow}\mathcal{A}\times\mathcal{B}$ and we want to calculate $\sum_{f\in\mathcal{F}} P(f)$, the sum of some property $P:\mathcal{F}\to\mathcal{Y}$ over all $f\in\mathcal{F}$.

If there exists $\alpha:\mathcal{A}\to\mathcal{Y}$ and $\beta:\mathcal{B}\to\mathcal{Y}$ such that for all $(a,b)\in(\mathcal{A}\times\mathcal{B})$

$$P\big(\sigma^{-1}(a,b)\big) = \alpha(a)\cdot\beta(b)$$

then the sum can be expressed as

$$\sum_{f\in\mathcal{F}} P(f) = \sum_{a\in\mathcal{A},b\in\mathcal{B}} P\big(\sigma^{-1}(a,b)\big) = \left(\sum_{a\in\mathcal{A}}\alpha(a)\right)\cdot\left(\sum_{b\in\mathcal{B}}\beta(b)\right)$$

In our case, $\mathcal{F}=\mathcal{Y}^{\mathcal{X}}$, $\mathcal{A}=\mathcal{Y}^{\mathcal{X}\setminus d^x_m}$ and $\mathcal{B}=\mathcal{Y}^{d^x_m}$.

(b) It only concerns algorithms that do not revisit previous points. This is not the case for many black box methods. Also, this limitation implies that the result expressed in theorem 5.2.1 only holds for $m \leq |\mathcal{X}|$. The result can be extended to algorithms that do revisit points, but do not get stuck in any set of points, so even considering the possibility of revisiting points, many gradient based optimization algorithms do not fulfill theorem 5.2.1's hypothesis.

(c) Some well known and widely used optimization algorithms are not considered, such as branch and bound types. This algorithms are not included because they use the cost of partial solutions to decide their paths.

In fact, we believe that (a) explains why this result is so unintuitive at first glance.

5.3 Other Results and Generalizations

Similar results exist for more specific optimization algorithms, like the ones regarding supervised learning algorithms [36–38] using Extended Bayesian Formalism (EBF). A very detailed presentation of EBF and its connection to supervised learning can be found in [36, Appendix A].

NFL theorems are true under the assumption of homogeneity in the cost functions. In [40, Sec. IV] a geometric interpretation of the theorems is given that allows to set a priori distinctions between learning algorithms based on properties of the cost function. Some specific problems may have cost functions with characteristic properties, such as the compositional structure of a neural network, that may suggest better performance of some optimization methods over the others.

Another interesting development is the continuous versions of NFL theorems presented in the previous section. We have not found in the literature a comprehensive generalization, but some work has been done [5, 26], stating that in some cases it can be generalized.

Chapter 6

Convergence of Stochastic Gradient Descent

6.1 Convergence of SGD to Global Minima: Open Problem

Given a network $\Sigma : \mathbb{R}^n \to \mathbb{R}$ with M trainable parameters and a collection of ν examples $\boldsymbol{z}_i \stackrel{\text{def}}{=} (\boldsymbol{x}^{(i)}, y^{(i)})$ with $i = 1 : \nu$ a ***loss function*** is a function $\mathcal{L} : \mathbb{R}^M \to \mathbb{R}$ that involves the examples and is to be minimized. There are many used loss function, being the square loss function the most common one:

$$\mathcal{L}_2(\boldsymbol{W}) = \frac{1}{\nu} \sum_{i=0}^{\nu} \left\| \Sigma(\boldsymbol{x}^{(i)}; \boldsymbol{W}) - y^{(i)} \right\|_2^2 \tag{6.1.1}$$

A typical technique for minimization of functions is gradient descent. This algorithm uses the idea that the gradient of a functions gives the direction of maximum increase, so minus the gradient gives the direction of maximum descent. A typical gradient descent algorithm is an iterative algorithm, that in each iteration:

$$\boldsymbol{W}^{k+1} = \boldsymbol{W}^k - \alpha_k \nabla \mathcal{L}(\boldsymbol{W}^k) \tag{6.1.2}$$

where learning rates α_k constitute a suitable sequence of positive numbers, that can be tuned to assure convergence.

In a general perspective, gradient descent methods converge to critical points, that is, points whose gradient vanishes. Critical points of a function f can be classified according to the eigenvalues of its Hessian matrix - the matrix containing its second derivatives - in the following way:

- Local maximum. When all eigenvalues are negative.
- Local minimum. When all eigenvalues are positive.

- Saddle points. When there are non-positive and non-negative eigenvalues.

Given some conditions, gradient descent methods generally converge to local minima. But then local minima can be classified into those that are actually ***global minima*** and those that are not, which we call ***bad local minima***. In principle, a gradient descent method is not able to distinguish between local and global minima, so this will be an important part of our analysis.

Gradient descent implies computing the gradient in each step, which is computationally expensive. Since the function to minimize in neural networks, the loss function, is generally computed as an average over all examples, a reasonable approach is to take the average on smaller sets of examples each time, called batches. To do so, the training set is randomly shuffled and divided into batches. Then at each iteration the gradient is computed using one batch, making it computationally cheaper, while adding some noise because the gradient is not computed to its full precision. This is called ***Stochastic Gradient Descent*** (SGD).

Intuition tells us that SGD may need more steps than regular GD to converge, but each step is computationally cheaper. Also, thanks to its randomness, SGD may be able to avoid bad local minima. Experiments show that in case of neural networks, this trade off eventually favors SGD.

Another useful variation that adds even more randomness to the algorithm is the following update rule:

$$\boldsymbol{W}^{k+1} = \boldsymbol{W}^k - \alpha_k \tilde{\nabla} \mathcal{L}(\boldsymbol{W}^k) + \gamma^k \boldsymbol{G}^k \tag{6.1.3}$$

where $\boldsymbol{G}^k$ is some white noise and γ^k is a sequence going to zero. This is introduced in [42] as ***Langevin Stochastic Gradient Descent*** (SGDL).

In this chapter we will prove that:

1. Batch gradient descent converges to local minima with probability 1 given a random parameter initialization.
2. SGD and SGDL converge to local minima provided the approximation of the gradient is *good enough* (we will give explicit details on what this means).
3. For multilayer neural networks, all differentiable local minima are in fact global minima.

These three points together give a set of rather mild conditions in which SGD as is used in real applications converges to a global minima. There are interesting cases that do not satisfy these conditions, but a comprehensive analysis of the problem cannot be found in the literature, as far as we know. Each result has its own conditions which it holds. We will give a summarized review of them and comment on their implications at the end of the chapter.

Another important topic related with optimization is the overall structure of the function: its ***landscape***. It is known that loss functions resulting from deep networks are non-convex, having possibly many bad

local minima and saddle points. [29] and [42] make an attempt to characterize this landscape, that we will explain in the last section of this chapter.

6.2 Convergence of Batch Gradient Descent to Local Minima

In this section we study the convergence of Batch Gradient Descent, that is standard gradient descent, assuming the only randomness comes from the initial value. The main result, described in [23] states that BGD converges almost surely with a random initialization and sufficiently small constant learning rate, provided the loss function is twice continuously differentiable and has no *non-strict saddles* (we will give precise definitions later).

The intuition behind this result is the following:

The main problem that can be found is a saddle point in which the algorithm gets stuck. Considering a saddle point, it has positive eigenvalues (representing directions of growth for the cost function), negative eigenvalues (representing directions of descent for the cost function) and null eigenvalues. When the direction chosen by the algorithm is in the span of the null eigenvectors, then there will be no effective descent in the cost function. If all saddles are strict (meaning that there is at least a negative eigenvalue) the span of nonnegative eigenvectors is a linear space of strictly less dimension than the whole vector space. Thus it has measure zero, which means that the probability of a direction being chosen in that particular space is zero. See [23, Sec. 3] for a more extensive explanation, with concrete examples.

6.2.1 Preliminary definitions and notation

Let $f : \Omega \subseteq \mathbb{R}^M \to \mathbb{R}$ be the function to minimize, with some domain Ω and that we will always suppose to be of class $\mathcal{C}^2(\Omega)$. For a given step size α, let $g : \Omega \subseteq \mathbb{R}^M \to \mathbb{R}^M$ be the ***gradient map***, that is:

$$g(x) = x - \alpha \nabla f(x) \tag{6.2.1}$$

Points of convergence of BGD correspond to critical points of f, that are exactly fixed points of g.

Definition 6.2.1 (L-smooth function). A function $f : \Omega \subseteq \mathbb{R}^M \to \mathbb{R}$ is said to be ***L-smooth*** if it is continuously differentiable and ∇f is L-Lipschitz:

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\| \tag{6.2.2}$$

We will assume that f is L-smooth for some constant L.

Since all norms in $\mathbb{R}^M$ are equivalent, the results will hold independent of the norm chosen.

Definition 6.2.2 (Iterates of gradient descent). The k-composition of g at an initial point x_0, denoted by $g^k(x_0)$ is the result of applying k iterations of the BGD algorithm. We will therefore denote the ***iterates*** as $x_k \stackrel{\text{def}}{=} g^k(x_0)$.

Definition 6.2.3 (Strict Saddle). A critical point x^* is a strict saddle if the minimum eigenvalue of the Hessian matrix is negative: $\lambda_{\min}\left(\nabla^2 f(x^*)\right) < 0$

Definition 6.2.4 (Global Stable Set). The ***global stable set*** of the critical point x^* is the set of initial conditions for which BGD converges to x^*:

$$W^s(x^*) = \{x \,:\, \lim_{k\to\inf} g^k(x) = x^*\} \tag{6.2.3}$$

We want to remark that a local maximum fits in our definition of strict saddle. We will prove that the probability of convergence to any kind of strict saddle is zero.

Definition 6.2.5 (Sublevel sets). Given a function $f : \Omega \subseteq \mathbb{R}^M \to \mathbb{R}$, its ***sublevel sets*** are the sets defined as

$$\{x \,:\, f(x) \leq C\} \tag{6.2.4}$$

where C is a given constant value.

We will be interested in the case that sublevel sets are compact. This is a very general condition for loss functions in machine learning. For example all functions satisfying $\lim_{\|x\|\to\infty} f(x) = \infty$ have compact sublevel sets.

Definition 6.2.6. Let C be the set of strict saddle points of f and let C^* be the set of local minima.

Definition 6.2.7. Let E be a vector space, and let $F_1, \ldots, F_k$ be subspaces of E. Consider the subspace F, the sum of the previous ones, defined as:

$$F = F_1 + \cdots + F_k \stackrel{\text{def}}{=} \{v_1 + \cdots + v_k \,:\, v_i \in F_i\} \tag{6.2.5}$$

If for any given vector $u \in F$, there exists only one array of vectors $(v_1, \ldots, v_k)$ such that $v_i \in F_i$ for all i and $u = v_1 + \cdots + v_k$, then we say that the sum is ***direct*** and we note it as $F = F_1 \oplus \cdots \oplus F_k$

6.2.2 Main results

For all results we will suppose the following conditions on f:

(a) f is twice continuously differentiable.

(b) f is L-smooth.

(c) f has no non-strict saddle.

Apart from these, many results will require extra conditions, that will be specified each time.

With all these definitions we can state the main theorem of this section.

Theorem 6.2.1 (Convergence of BGD). *Let f be a function with countably many critical points and $\lim_k x_k$ exists for all initial value x_0. Then $Pr(\lim_k x_k \in C^*) = 1$.*

In this theorem the assumption that seems non-trivial is the existence of the limit $\lim_k x_k$. To justify this assumption we have the following result:

Proposition 6.2.2 (Proposition 12.4.4 of [22]). *Assume f has isolated critical points and compact sublevel sets, then $\lim_k x_k$ exists.*

This condition in itself is fairly general for machine learning purposes. Nevertheless, at the end of the section we will give another sufficient condition, based on Łojasiewicz inequality, see definition 6.2.8.

6.2.3 Proof of the results

The main tool we will use is the Stable-Center Manifold theorem, presented in [33] and first developed in [34**?**]. The formal statement of the theorem is the following:

Theorem 6.2.3 (Stable-Center Manifold, Theorem III.7 in [33]). *Let $U \subseteq \mathbb{R}^M$ be a neighborhood of a point $p \in \mathbb{R}^M$. Let $\varphi : U \to \mathbb{R}^M$ be a local diffeomorphism with p as a fixed point. Consider the decomposition $\mathbb{R}^M = E_s \oplus E_u$, where E_u (unstable) is the span of the generalized eigenvectors of $D\varphi(p)$ whose corresponding eigenvalue is greater than one and E_s (stable) is the span of the rest of its generalized eigenvectors. Then there exists an embedded disk W_{loc} that is tangent to E_s at p, called the **local stable center manifold**. In addition, there exists B, a neighborhood of p such that $\varphi(W_{loc}) \cap B \subseteq W_{loc}$ and if $x \in \varphi^k(B)$ for all k, then $x \in W_{loc}$.*

This theorem is saying that given a diffeomorphism φ and a splitting of $\mathbb{R}^M$ according to the eigenvalues of $D\varphi(0)$, there exists a manifold W_{loc} containing 0, of dimension the same as E_s, that contains all points converging to 0. For our purposes this will mean that all points converging to a given one (in this case 0) lay in a manifold of a given dimension. When this dimension is less than M (full dimension of the environment space) then the stated manifold will have zero measure.

We will apply this theorem using g as our diffeomorphism. To do so, we will need the following result:

Lemma 6.2.4. *The gradient mapping g is a diffeomorphism, provided the step size α is less than $1/L$.*

PROOF.
First observe that g is continuously differentiable by hypothesis, since $f \in \mathcal{C}^2$.

To prove the function is bijective we will explicitly build its inverse function. Consider the auxiliary function $\gamma_y(x) = \frac{1}{2}\|x - y\| - \alpha f(x)$. This function is strongly convex for $\alpha < 1/L$, so it has a unique minimizer. Let x_y be such minimizer. By KKT conditions [17, 20] we have that

$$y = x_y - \nabla f(x_y) = g(x_y) \tag{6.2.6}$$

This tells us that given $g(x_y)$, we can find a x_y the minimizer of $\gamma_y(x)$, so x_y as a function of y is the inverse function.

We have to prove now that g^{-1} is continuously differentiable. We observe that the condition $\alpha < 1/L$ implies that $Dg(x)$ is and invertible matrix, so we can apply the inverse theorem to g to prove g^{-1} is continuously differentiable. ■

Now let us prove an important result that states "non-convergence to saddle points":

Theorem 6.2.5 (Theorem 4.1 in [23]). *Let x^* be an strict saddle critical point, and $\alpha < 1/L$. Then the probability of reaching it is zero:* $\Pr(\lim_k x_k = x^*) = 0$

PROOF.
Observe that $Dg(x) = I - \alpha\nabla^2 f(x)$. Therefore the dimension of W_{loc} is the number of non-negative eigenvalues of $\nabla^2 f(x)$. By the assumption of f not having non-strict saddle this dimension is strictly less than the environment dimension, so it will have measure zero.

Consider x^* a critical point of f (and thus a fixed point of g). Let B be the neighborhood of x^* as in theorem 6.2.3. If an the gradient iterates with initial point x_0 converge to x^*, there must exist some T such that $g^t(x_0) \in B$ for all $t \geq T$. This means that $g^k(x_T) \in B$ for all k, so by theorem 6.2.3 $x_T \in W_{loc}$. Thus, regarding the global stable set of x^*, we have shown that:

$$W^s(x^*) \subseteq \bigcup_{T=0}^{\infty} g^{-T}\left(W_{loc}\right) \tag{6.2.7}$$

Since g is a diffeomorphism, so are the g^T's. Diffeomorphisms map sets of measure zero to sets of mesure zero, and the countable union of sets of measure zero is of measure zero, so $W^s(x^*)$ has measure zero, equivalently $\Pr(\lim_k x_k = x^*) = 0$. ■

Corollary 6.2.6. *If the set C of strict saddle points of f has at most countably infinite cardinality then* $\Pr(\lim_k x_k \in C) = 0$.

PROOF.
The proof consists on applying the same reasoning as in theorem 6.2.5 to each critical point. The union of countably many sets of measure zero has still measure zero. ■

This cardinality condition must not be considered a restriction. No reasonable functions will have uncountably many saddle points. In fact, we will demand saddle points to be isolated.

Now the proof of theorem 6.2.1 is direct. We know that whenever there exists $\lim_k x_k$ it must be a critical point. If the probability of reaching a strict saddle is null, and these are the only critical points that are not local minima, then if there is convergence, $\Pr(\lim_k x_k \in C^*) = 1$.

6.2.3.1 Existence of limit

Proposition 6.2.2 is presented in [22, Prop. 12.4.4] in the context of MM algorithms. Indeed, BGD with fixed step $\alpha < 1/L$ is an example of MM algorithm. We have adapted the proofs so that we do not need to talk explicitly of this kind of algorithms.

We will need three previous lemmas before we can attack the proof of proposition 6.2.2.

Lemma 6.2.7. *Let f be a function with L-Lipschitz gradient. Then for all x, y in its domain it is satisfied:*

$$f(x) \leq f(y) + \nabla f(y)^T (x - y) + \frac{L}{2}\|x - y\|_2^2 \tag{6.2.8}$$

PROOF.
Consider the auxiliary function $h(t) \stackrel{\text{def}}{=} f\big(y + t(x - y)\big)$. By construction, $h(1) = f(x)$, $h(0) = f(y)$ and

$$h'(t) = \nabla f\big(y + t(x - y)\big)^T (x - y) \tag{6.2.9}$$

We also remark that $\int_0^1 h'(t)dt = h(1) - h(0) = f(x) - f(y)$. Then:

$$\Big|f(x) - f(y) - \nabla f(y)^T (x - y)\Big| = \left|\int_0^1 \nabla f\big(y + t(x - y)\big)^T (x - y)dt - \int_0^1 \nabla f(y)^T (x - y)dt\right| \tag{6.2.10}$$

$$\leq \int_0^1 \Big|\big[\nabla f\big(y + t(x - y)\big) - \nabla f(y)\big]^T (x - y)\Big| \, dt \tag{6.2.11}$$

Now applying Cauchy-Schwartz inequality and ∇f being L-Lipschitz:

$$\leq \int_0^1 \big\|\nabla f\big(y + t(x - y)\big) - \nabla f(y)\big\| \, \|(x - y)\| \, dt \tag{6.2.12}$$

$$\leq \|x - y\| \int_0^1 Lt\|x - y\|dt = \frac{L}{2}\|x - y\|^2 \tag{6.2.13}$$

We have shown that

$$\left|f(x) - f(y) - \nabla f(y)^T(x-y)\right| \leq \frac{L}{2}\|x-y\|_2^2 \tag{6.2.14}$$

Now applying the so called reverse triangular inequality to the LHS the proof is done. ■

Lemma 6.2.8. *The sequence of costs of the iterates satisfies:*

$$f(x_{k+1}) \leq f(x_k) \tag{6.2.15}$$

and the equality only happens when $x_{k+1} = x_k$ is a critical point.

PROOF.
Consider the auxiliary function $h_\alpha(x,y) \stackrel{\text{def}}{=} f(y) + \nabla f(y)^T(x-y) + \frac{1}{2\alpha}\|x-y\|_2^2$. We will show that

$$x_{k+1} = \arg\min_x h_\alpha(x, x_k) \tag{6.2.16}$$

Since f is L-Lipschitz, it is also $\tilde{L}$-Lipschitz for any $\tilde{L} \geq L$, in particular it is $1/\alpha$-Lipschitz. As a consequence, lemma 6.2.7 implies

$$f(x) \leq h_\alpha(x,y) \quad \text{for all } y \tag{6.2.17}$$

Assuming eq. (6.2.16), the following inequalities hold:

$$f(x_{k+1}) \leq h_\alpha(x_{k+1}, x_k) \leq h_\alpha(x_k, x_k) = f(x_k) \tag{6.2.18}$$

As we will see, the second inequality is strict whenever $x_{k+1} \neq x_k$. Let's prove now eq. (6.2.16).

Since h_α is a convex quadratic function with respect to x, the only minimizer x^* is the one that fulfills $\nabla_x h_\alpha(x^*, y) = 0$. In particular, for any $x \neq x^*$, $h_\alpha(x^*, y) < h_\alpha(x, y)$. Differentiating h_α with respect to x we obtain:

$$\nabla_x h_\alpha(x,y) = \nabla f(y) - \frac{1}{\alpha}(x-y) \tag{6.2.19}$$

Thus the minimizer is $x^* = y - \alpha\nabla f(y)$, proving eq. (6.2.16). ■

Lemma 6.2.9. *If a bounded sequence $\{y_k\}_{k\in\mathbb{N}}$ in $\mathbb{R}^M$ satisfies*

$$\lim_k \|y_{k+1} - y_k\| = 0 \tag{6.2.20}$$

then its set T of cluster points defined as

$$T \stackrel{def}{=} \{y \,:\, \lim_k y_{m_k} = y \text{ for some subsequence } \{y_{m_k}\}_k\} \tag{6.2.21}$$

is connected. And if it is finite, then it has only one point $y = \lim_k y_k$.

PROOF.
The limit of cluster points is a cluster point, so T is closed. Since the sequence is bounded, T is also bounded, and therefore compact. As usual, we will prove connection by contradiction. Suppose T is not connected. Then there exist two closed sets C_0 and C_1 that disconnect T. Let $T_0 \stackrel{\text{def}}{=} T \cap C_0$ and $T_1 \stackrel{\text{def}}{=} T \cap C_1$. Then T_0 and T_1 are both nonempty and disjoint because C_0 and C_1 disconnect T. They are also both compact because they are a closed subset of the compact T. Since they are compact and disjoint, the distance, defined as:

$$\text{dist}(T_0, T_1) \stackrel{\text{def}}{=} \inf_{u_0 \in T_0} \text{dist}(u_0, T_1) \stackrel{\text{def}}{=} \inf_{u_i \in T_i} \|u_0 - u_1\| \tag{6.2.22}$$

is strictly positive. Let δ be such distance. Now let's consider the sequence $\{y_k\}$. Equation (6.2.20) implies the existence of K such that for all $k \geq K$,

$$\|y_{k+1} - y_k\| < \delta/4 \tag{6.2.23}$$

Since both T_0 and T_1 are nonempty and contain cluster points of $\{y_k\}$, the elements of the sequence must "travel" infinitely many times from T_0 to T_1 and the other way back, entering an infinite amount of times in the closed region $W \stackrel{\text{def}}{=} \{u : \text{dist}(u, T) \geq \delta/4\}$. But in that case, W would contain a cluster point, and by construction $W \cap T = \emptyset$. The contradiction comes from the assumption of T being disconnected.

If a set is finite and connected, it can only be a singleton or the empty set. The set of cluster points cannot be empty because the sequence is bounded, and therefore it contains some converging subsequence, whose limit is a cluster point. ■

PROOF. (Of proposition 6.2.2)
Consider an initial point x_0 and its iterates sequence $\{x_k\}_{k \in \mathbb{N}}$. Let Γ be the set of cluster points of $\{x_k\}$. We want to prove that Γ has exactly one element. To do so, we will prove the following results:

i) $\Gamma \subseteq S$, where S is the set of fixed points of g, and S is finite.

ii) Γ satisfies eq. (6.2.20) of lemma 6.2.9.

Then $\Gamma \subseteq S$ implies Γ finite, and lemma 6.2.9 implies Γ has exactly one element.

$\boxed{(i)}$ Consider a cluster point $z \in \Gamma$. There exists a subsequence $\{x_{m_k}\}_k$ such that $\lim_k x_{m_k} = z$. The sequence of costs $\{f(x_k)\}_k$ is monotonically decreasing by lemma 6.2.8 and bounded below by the hypothesis of f having compact sublevel sets. Therefore $\lim_k f(x_k)$ exists. For all $k \in \mathbb{N}$, the following inequalities hold:

$$f\left(x_{m_{(k+1)}}\right) \leq f\left(x_{m_k+1}\right) = f(g(x_{m_k})) \leq f(x_{m_k}) \tag{6.2.24}$$

Since we know that both the limits of the LHS and RHS exist, we can take limits in the previous chain of inequalities, and by the sandwich rule, $f(g(z)) = f(z)$. In principle f is not an injective function, but

lemma 6.2.8 tells us [1] that this equality only happens when z is a critical point of f, equivalently a fixed point of g.

$\boxed{(ii)}$ We will follow by contradiction. Since $\{x_k\}_k$ lies inside the compact set $\{x \,:\, f(x) \leq f(x_0)\}$, we can extract a subsequence $\{x_{m_k}\}_k$ converging to some limit $u = \lim_k x_{m_k}$. Now consider the sequence $\{z_k\}_k = \{x_{m_k+1}\}_k$. Again, it lies inside a compact set, so there is a subsequence $\{z_{n_k}\}_k$ converging to some limit $v = \lim_k z_{m_k}$. If we suppose eq. (6.2.20) does not hold for $\{x_k\}_k$, then $u \neq v$. But applying continuity of g, $g(u) = v$, so u would not be a fixed point of g, contradicting (i). ■

6.2.4 Bounds on convergence rates

As in [23] we will give some bounds on the convergence rates. To do so, we need to introduce the Łojasiewicz gradient inequality, which characterizes the steepness of the gradient in the neighborhood of a critical point. We will also use this inequality to provide another sufficient condition, independent from proposition 6.2.2, to ensure the existence of a limit for the sequence of iterates.

We will skip the proofs, that can be found in [2] and [23].

Definition 6.2.8 (Łojasiewicz Gradient Inequality). A critical point x^* satisfies Łojasiewicz Gradient Inequality with parameters $a \in [0, 1)$ and $m > 0$ if there exists a neighborhood $U \ni x^*$ and $\varepsilon > 0$ such that

$$\|\nabla f(x)\| \geq m\,|f(x) - f(x^*)|^a \tag{6.2.25}$$

for all $x \in \{x \in U \,:\, f(x^*) < f(x) < f(x^*) + \varepsilon\}$

The Łojasiewicz inequality is quite general. As an example, every real analytic function satisfies it for some value of the parameters. A more extensive discussion can be found in [8].

Proposition 6.2.10. *Consider the same conditions as in corollary 6.2.6, plus the assumption of f satisfying the Łojasiewicz inequality for some value of the parameters a and m and the sequence of iterates $\{x_k\}_k$ being bounded. Then $\lim_k x_k$ exists.*

Proposition 6.2.11. *With the same conditions of proposition 6.2.10, there exist some C and b independent of k such that:*

1. *If $a \in (0, 1/2)$, then: $\|x_k - x^*\| \leq Cb^k$*
2. *If $a \in (1/2, 1)$, then; $\|x_k - x^*\| \leq \dfrac{C}{k^{(1-a)/(2a-1)}}$*

[1] If one substitutes x_k in eq. (6.2.15) by z and x_{k+1} by $g(z)$ this implication becomes clearer.

6.3 Convergence of Stochastic Gradient Descent to Local Minima

In this section we will present, without proof, results mainly from [12] proving that SGD and SGDL converge to local minima almost everywhere under certain conditions.

Definition 6.3.1 (Gradient Oracle). Let $\mathscr{X}$ be the set of random vectors in $\mathbb{R}^n$. For a function $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$, a random function $SG : \Omega \to \mathscr{X}$ is a ***gradient oracle*** of radius $Q > 0$ if for all $w \in \Omega$

$$\mathbb{E}[SG(w))] = \nabla f(w) \qquad \text{and} \qquad \|SG(w) - \nabla f(w)\| \leq Q \tag{6.3.1}$$

We will also use the concept of strict saddle, but in this section we will give a robust version of the property of a function not having any non-strict saddles

Definition 6.3.2 ($(\alpha, \gamma, \epsilon, \delta)$-Strict Saddle). Given $\alpha, \gamma, \epsilon, \delta > 0$, a function $f \in \mathcal{C}^2(\Omega)$ is a ***$(\alpha, \gamma, \epsilon, \delta)$-strict saddle*** if for any point $w \in \Omega$ at least one of the following conditions hold:

1. $\|\nabla f\| \geq \epsilon$.
2. $\lambda_{\min}(\nabla^2 f(w)) < -\gamma$.
3. There is a local minimum w^* such that $\|w - w^*\| \leq \delta$, and the function $f|_W$ is α-strongly convex, where $W \stackrel{\text{def}}{=} \{w' : \|w' - w^*\| \leq 2\delta\}$.

The main theorem of this section, which corresponds to [12, Th. 6], gives conditions for SGD to converge with arbitrary high probability to a point arbitrary close to a local minimum. In a practical setup, one could use SGD until theorem 6.3.1 ensures the iterates enter a region where the loss function is strongly convex, and there apply suitable methods for convex optimization.

For theorem 6.3.1 to hold, as can be seen in its proof, we need enough variance in the gradient to be able to escape saddle points. We will suppose the aforementioned condition on the gradient oracle. If unsure of the gradient oracle's variance, one can always modify the oracle adding a random variable that is uniform in the sphere of radius 1, as in [12, Algorithm 1]. In the worst case this increases the radius of the oracle from Q to $Q + 1$.

Theorem 6.3.1 (Convergence of SGD). *Let $f : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}$ be an $(\alpha, \gamma, \epsilon, \delta)$-strict saddle, bounded by $|f(w)| \leq B$, β-smooth and with ρ-Lipschitz Hessian. Let SG be a gradient oracle of f with radius Q. Then there exists $\eta_{\max} = \Theta(1)$ such that for all $\zeta \in (0, e^{-1})$ and for any $\eta < \log(1/\zeta)$, with probability at least $1-\zeta$ in $t = \mathcal{O}\left(\eta^{-2}\log(1/\zeta)\right)$ iterations of SGD, the iterate w_t is $\mathcal{O}(\sqrt{\eta \log(1/\eta\zeta)})$-close to some local minimum w^*.*

In this theorem asymptotic notation $\Theta(\cdot), \mathcal{O}(\cdot)$ only takes into account the dependence on η and θ, hiding all factors that depend on $Q, \alpha, \gamma, \epsilon, \delta, B, \beta, \rho$ and d, but are independent of η and θ.

The proof of this theorem is long and uses advanced probability concepts, so we will skip it. The interested reader is referred to [12].

6.4 Local Minima are Global Minima for Certain Networks

In this section we present a result first appeared in [35] proving that in a certain kind of neural networks, all differentiable local minima are in fact global minima.

As in the previous section, this is only a partial result, but as far as we know, there is no complete answer available at the moment of writing.

We will skip the proofs, that can be found in [35].

The results presented in this section have three main constraints, being the restriction on the activation functions the main one, because it sets a strong limitations on which networks these results apply:

(i) Activation functions are restricted.

(ii) Gaussian dropout-like noise is required for the results to hold.

(iii) The result only applies to differentiable local minima, leaving the question open for non-differentiable ones.

We will follow a similar matrix notation as in [35].

6.4.1 Definitions and Notation

Consider a Multilayer Neural Network (MNN) of L layers. This network is trained with N examples $\{\boldsymbol{x}^{(n)}, y^{(n)}\}_{n=1:N}$. Let $\boldsymbol{X} \overset{\text{def}}{=} [\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}] \in \mathbb{R}^{d_0 L}$ and $\boldsymbol{y} = [y^{(1)}, \dots y^{(N)}] \in \mathbb{R}^N$. In each layer l the layer inputs $\boldsymbol{u}_l^{(n)}$ and layer outputs $\boldsymbol{v}_l^{(n)}$ are defined by:

$$\boldsymbol{u}_l^{(n)} \overset{\text{def}}{=} \boldsymbol{W}_l \boldsymbol{v}_{l-1}^{(n)} \qquad \text{and} \qquad \boldsymbol{v}_l^{(n)} \overset{\text{def}}{=} \operatorname{diag}\left(\boldsymbol{a}_l^{(n)}\right) \boldsymbol{u}_l^{(n)} = \begin{bmatrix} a_{1,l}^{(n)}\left(\boldsymbol{u}_l^{(n)}\right) \\ \vdots \\ a_{d_l,l}^{(n)}\left(\boldsymbol{u}_l^{(n)}\right) \end{bmatrix} \tag{6.4.1}$$

where the input of the network is $\boldsymbol{v}_l^{(n)} \overset{\text{def}}{=} \boldsymbol{x}^{(n)}$, matrices $\boldsymbol{W}_l$ represent the weights and $\boldsymbol{a}_l^{(n)}$ are the activations. We will restrict the activations to have the following structure:

$$a_L^{(n)} = 1, \quad \forall l < L: \quad a_{i,l}^{(n)}(\boldsymbol{u}) \overset{\text{def}}{=} \varepsilon_{i,l}^{(n)} \cdot \begin{cases} 1 & \text{if } u_i \geq 0 \\ s & \text{if } u_i < 0 \end{cases} \tag{6.4.2}$$

Where s is some slope and $\boldsymbol{\mathcal{E}}_l \overset{\text{def}}{=} \left[\boldsymbol{\varepsilon}_l^{(1)}, \dots \boldsymbol{\varepsilon}_l^{(N)}\right]$ is a random variable, that acts a dropout-like noise. Observe that if $\boldsymbol{\mathcal{E}}_l$ is a deterministic matrix of ones we recover the classical leaky ReLU, and if all $\varepsilon_{i,l}^{(n)}$ are distributed in the discrete set $\{0, 1\}$ we they represent a realization of dropout noise. We will study the case of $(\boldsymbol{\mathcal{E}}_1, \dots, \boldsymbol{\mathcal{E}}_{L-1})$ are i.i.d. Gaussian entries and $\boldsymbol{X}$ is smoothed with arbitrary small Gaussian noise.

Let $\boldsymbol{e} \overset{\text{def}}{=} \boldsymbol{v}_L - \boldsymbol{y}$ be the output error. The loss function to minimize is the mean square error defined in terms of the empirical expectation $\hat{\mathbb{E}}$ as:

$$\text{MSE} \overset{\text{def}}{=} \frac{1}{2}\hat{\mathbb{E}}\boldsymbol{e}^2 \overset{\text{def}}{=} \frac{1}{2N}\boldsymbol{e}^T\boldsymbol{e} \tag{6.4.3}$$

Definition 6.4.1 (Differentiable Local Minimum). A minimum is said to be differentiable if for all i, l, n the input satisfies $u_{i,l}^{(n)} \neq 0$.

We will be studying the training error, i.e. the MSE on differentiable local minima.

With these definitions, we can present two results:

Theorem 6.4.1 (Th. 4 in [35]). *For a single hidden layer network ($L = 2$), if $N \leq d_1 d_0$, then all differentiable local minima of MSE (eq.* (6.4.3)*) are global minima with $MSE = 0$, $(\boldsymbol{X}, \boldsymbol{\mathcal{E}}_1)$ almost everywhere.*

Theorem 6.4.2 (Th. 5 in [35]). *For a multiple hidden layer network ($L > 2$), if $N \leq d_{L-2} d_{L-1}$, then all differentiable local minima of MSE with respect to $\boldsymbol{W}_{L-1}$ and $\boldsymbol{W}_L$, considering $\boldsymbol{W}_1, \dots \boldsymbol{W}_{L-2}$ fixed, are global minima with $MSE = 0$, $(\boldsymbol{X}, \boldsymbol{\mathcal{E}}_1, \dots, \boldsymbol{\mathcal{E}}_{L-1}, \boldsymbol{W}_1, \dots, \boldsymbol{W}_{L-2})$ almost everywhere.*

6.5 Landscape of Loss Function

So far the approach to the problem of convergence has been to constrain the problem and then give answers to the constrained problem, so that constitute different answer for different special cases.

Another approach, more ambitious than the former ones, developed by Poggio et al. in [29, 42], is give a detailed description of the *landscape* of the loss function, so that properties can be much better understood. In this section we will briefly develop the aforementioned approach.

6.5.1 Hyper Basin Model

Based on experimental data presented in [29], the following observations can be made:

(i) A small perturbation leads to a slightly different convergence path. The earlier in the training process the perturbation, the more different the paths would be.

(ii) Interpolation of two nearby converging paths leads to another converging path. Interpolation of two distant converging path does not lead, in general, to a converging path.

(iii) No bad local minima are observed.

With all these properties, Poggio et al. propose two models based on the concept of ***hyper basin***, i.e. basin in a high number of dimensions.

(a) ***Basin model:*** In this model the landscape is just a collection of hyper basins, each around a flat global minima.

(b) ***Basin-fractal model:*** In this model the landscape is composed of hyper basins that have other hyper basins inside, in a fractal disposition. Even though this models seems to be more elegant, it would require the existence of "walls" between similar models (those that are in the same basin at some fractal level, but are in different basins in deeper levels), that are not found in experimental results presented in [29].

6.5.2 Langevin Equation Model

In [42] the problem is studied in a probabilistic approach, by considering it an approximate Langevin equation (used in statistical mechanics to describe, for example, Brownian motion).

Consider that ν examples are drawn i.i.d. from a probability space Z with some probability measure ρ. In this case, the loss function $\mathcal{L}$ can be regarded as $\mathcal{L}(\boldsymbol{W}, \boldsymbol{z})$, a function of the weights and also of the random variable $\boldsymbol{z}$. Assume that for all $\boldsymbol{W}$, the expected loss exists:

$$I(\boldsymbol{W}) \stackrel{\text{def}}{=} \mathbb{E}_{\boldsymbol{z}} \mathcal{L}(\boldsymbol{W}, \boldsymbol{z}) \tag{6.5.1}$$

With this setup, we want to find the minimizer $\boldsymbol{W}^*$ such that

$$I(\boldsymbol{W}^*) = \min_{\boldsymbol{W} \in \mathbb{R}^M} I(\boldsymbol{W}) \tag{6.5.2}$$

For the structure of neural networks and the limitations of computers, we can consider that feasible $\boldsymbol{W}$'s are in a compact space $K \subseteq \mathbb{R}^M$.

In general, the minimizer $\boldsymbol{W}^*$ is not guaranteed to exist nor to be unique.

We can now rewrite eq. (6.1.2) as

$$\boldsymbol{W}^{k+1} = \boldsymbol{W}^k - \alpha_k \left(\nabla I(\boldsymbol{W}^k) + \boldsymbol{\xi}_k \right) \tag{6.5.3}$$

where $\boldsymbol{\xi}_k$ is a noise equivalent quantity defined as

$$\boldsymbol{\xi}_k \stackrel{\text{def}}{=} \nabla\mathcal{L}(\boldsymbol{W}^k; \boldsymbol{z}_k) - \nabla I(\boldsymbol{W}_k) \tag{6.5.4}$$

being $\boldsymbol{z}_k$ the mini-batch of examples used in step k to approximate the gradient. By definition the noise satisfies $\mathbb{E}(\boldsymbol{\xi}_k) = \mathbf{0}$.

Now eq. (6.5.3) is a discretization of a Langevin equation for diffusion, with α_k instead of $\sqrt{\alpha_k}$. In [42, Sec. 3] it is exposed how under certain conditions, the associated Langevin equation converges to a probability distribution that *prefers* degenerate (flat) minima over non-degenerate ones, and how flat minima will lead to more robust optimization.

6.6 Introduction to The Problem of Generalization

The big open question surrounding artificial neural network is generalization, defined as the performance of a model over unseen data. Generally, when trained, available data is divided in a training set, that is used to train the model, and a test set, that is used to test generalization.

It is also a significant issue in actual problems, because generalization directly affects the performance of learning algorithms. A model that fits the training data, but fails to fit in previously unseen data (typically due to overfitting) is essentially useless for practical purposes.

In this chapter we will summarize what we think are the most relevant contributions to this problem.

6.6.1 Regularization Techniques

Regularization techniques are modifications of the learning algorithms used to make the resulting model more *regular*, thus preventing overfitting. This is very useful to allow a model to use much more data than parameters, while preventing it to overfit to the extra data.

A collection of the most common regularization techniques can be found in [14, Ch. 7].

Although these modifications are designed and intended to reduce generalization error, and they do it in many practical cases, there is not much of a theoretical explanation of why this happens. In fact, some experiments [41] suggest that generalization cannot be explained by regularization methods.

6.6.2 Implicit Regularization

Some experiments [31, 41] suggest that there may be an implicit regularization effect in both the structure of the network and the SGD learning algorithm. The idea presented in [42] and summarized in our sec-

tion 6.5 of SGD being an approximate Langevin equation may lead to explain this implicit regularization effect of SGD.

6.6.3 Classical Generalization Bounds

There exist some classical results of general optimization regarding generalization properties, that do not apply to our neural networks, but may be a source of inspiration to find analogous results in our case. In [41, Appendix 8] there is a thorough exposition of these bounds and some of the problems when trying to apply them to neural networks.

Bibliography

[1] A. DeVore, R., Howard, R., and Micchelli, C. Optimal nonlinear approximation. *Manuscripta Mathematica 63* (Dec. 1989), 469–478.

[2] Absil, P., Mahony, R., and Andrews, B. Convergence of the Iterates of Descent Methods for Analytic Cost Functions. *SIAM Journal on Optimization 16*, 2 (Jan. 2005), 531–547.

[3] Adams, R., and Fournier, J. *Sobolev Spaces*. Pure and Applied Mathematics. Elsevier Science, 2003.

[4] Arnol'd, V. I. On the representation of continuous functions of three variables as superpositions of continuous functions of two variables. *Dokl. Akad. Nauk SSSR 114*, 4 (1957), 679–681.

[5] Auger, A., and Teytaud, O. Continuous Lunches Are Free! In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2007), GECCO '07, ACM, pp. 916–922.

[6] Bagby, T., Bos, L., and Levenberg, N. Multivariate simultaneous approximation. *Constructive Approximation 18*, 4 (Dec. 2002), 569–577.

[7] Baire, R. Sur les fonctions de variables réelles. *Annali di Matematica Pura ed Applicata (1898-1922) 3*, 1 (Dec. 1899), 1–123.

[8] Bolte, J., Daniilidis, A., Ley, O., and Mazet, L. Characterizations of Łojasiewicz inequalities: Subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society 362*, 6 (2010), 3319–3363.

[9] Burden, R., and Faires, J. *Numerical Analysis*. Cengage Learning, 2010.

[10] Corominas, E., and Balaguer, F. S. Condiciones para que una función infinitamente derivable sea un polinomio. *Revista matemática hispanoamericana 14*, 1 (1954), 26–43.

[11] Ebbinghaus, H. Ueber Das Gedächtnis. *Mind 10*, 39 (1885), 454–459.

[12] Ge, R., Huang, F., Jin, C., and Yuan, Y. Escaping From Saddle Points — Online Stochastic Gradient for Tensor Decomposition. *arXiv:1503.02101 [cs, math, stat]* (Mar. 2015).

[13] Girosi, F., and Poggio, T. Representation Properties of Networks: Kolmogorov's Theorem Is Irrelevant. *Neural Computation 1*, 4 (Dec. 1989), 465–469.

[14] Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

[15] Hörmander, L. *The Analysis of Linear Partial Differential Operators: Distribution Theory and Fourier Analysis*. Springer Study Edition. Springer-Verlag, 1990.

[16] Jackson, D. On Approximation by Trigonometric Sums and Polynomials. *Transactions of the American Mathematical Society 13*, 4 (1912), 491–515.

[17] Karush, W. Minima of functions of several variables with inequalities as side constraints. *M. Sc. Dissertation. Dept. of Mathematics, Univ. of Chicago* (1939).

[18] Kolmogorov, A. N. On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables. *Dokl. Akad. Nauk SSSR 108*, 2 (1956), 179–182.

[19] Kůrková, V. Kolmogorov's Theorem is Relevant. *Neural Comput. 3*, 4 (Dec. 1991), 617–622.

[20] Kuhn, H. W., and Tucker, A. W. Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950* (1951), University of California Press, Berkeley and Los Angeles, pp. 481–492.

[21] Lang, S. *Real and Functional Analysis*, 3 ed. Graduate Texts in Mathematics 142. Springer-Verlag New York, 1993.

[22] Lange, K. *Optimization*, 2 ed. Springer Texts in Statistics. Springer-Verlag, New York, 2013.

[23] Lee, J. D., Simchowitz, M., Jordan, M. I., and Recht, B. Gradient Descent Converges to Minimizers. *arXiv:1602.04915 [cs, math, stat]* (Feb. 2016).

[24] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks 6*, 6 (1993), 861 – 867.

[25] Lin, V. Y., and Pinkus, A. Fundamentality of Ridge Functions. *Journal of Approximation Theory 75*, 3 (1993), 295 – 311.

[26] Lockett, A. J., and Miikkulainen, R. A Probabilistic Reformulation of No Free Lunch: Continuous Lunches Are Not Free. *Evolutionary Computation 25*, 3 (Oct. 2016), 503–528.

[27] Mhaskar, H. N. Neural Networks for Optimal Approximation of Smooth and Analytic Functions. *Neural Computation 8* (1996), 164–177.

[28] Pinkus, A. Approximation theory of the MLP model in neural networks. *Acta Numerica 8* (1999), 143–195.

[29] Poggio, T., and Liao, Q. Memo 66: Theory II: Landscape of the Empirical Risk in Deep Learning. *Center for Brains, Minds and Machines* (2017).

[30] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. Memo 58: Why and When Can Deep - but Not Shallow - Networks Avoid the Curse of Dimensionality: A Review. *Center for Brains, Minds and Machines* (2016).

[31] Poggio, T., Zhang, C., Liao, Q., Rakhlin, A., Sridharan, K., Miranda, B., and Golowich, N. Memo 67: Theory of Deep Learning III: Generalization Properties of SGD. *Center for Brains, Minds and Machines* (2017).

[32] Salsa, S. *Partial Differential Equations in Action: From Modelling to Theory*. Universitext. Springer Milan, 2008.

[33] Shub, M. *Global Stability of Dynamical Systems*. Springer-Verlag, New York, 1987.

[34] Smale, S. Differentiable dynamical systems. *Bulletin of the American Mathematical Society 73*, 6 (1967), 747–817.

[35] Soudry, D., and Carmon, Y. No bad local minima: Data independent training error guarantees for multilayer neural networks. *ArXiv e-prints* (May 2016).

[36] Wolpert, D. H. The Existence of a Priori Distinctions Between Learning Algorithms. *Neural Computation 8*, 7 (Oct. 1996), 1391–1420.

[37] Wolpert, D. H. The Lack of A Priori Distinctions Between Learning Algorithms. *Neural Computation 8*, 7 (Oct. 1996), 1341–1390.

[38] Wolpert, D. H. The Supervised Learning No-Free-Lunch Theorems. In *Soft Computing and Industry*. Springer, London, 2002, pp. 25–42.

[39] Wolpert, D. H., and Macready, W. G. *No Free Lunch Theorems for Search*. Technical Report SFI-TR-95-02-010, Santa Fe Institute, 1995.

[40] Wolpert, D. H., and Macready, W. G. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation 1*, 1 (Apr. 1997), 67–82.

[41] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530 [cs]* (Nov. 2016).

[42] Zhang, C., Liao, Q., Rakhlin, A., Miranda, B., Golowich, N., and Poggio, T. Theory of Deep Learning IIb: Optimization Properties of SGD, 2017.

Appendices

Appendix A

Conventions of symbols

Symbol	Use
a	Otimization algorithm, $a : \mathcal{D} \to \mathcal{X}$
b	Bias(es)
$B(\boldsymbol{x}, r)$	Ball center x radius r
d	Depth (of a layer)
$\mathcal{D}$	Space of $\boldsymbol{d}$'s $(\mathcal{X}, \mathcal{Y})$
$\boldsymbol{d}$	Vector of examples
$D^{\boldsymbol{k}}F$	Derivative
$E()$	Best approx error, many ways
$E_k(f)$	Best approximation error of f by a poly of deg k
f	Target function (function to approximate)
f	Function, in general
f^*	Best approximation to f
g	Function, in general
I	Expected loss of $\mathcal{L}$
K	Compact
$\mathcal{C}^k$	Continuously differentiable spaces
$\mathcal{G}$	Graph, for $\mathcal{G}$-functions
$\mathcal{L}^p$	$\mathcal{L}^p$ spaces
$\mathcal{L}$	Loss function
m	Number of derivatives of the target function f
m	Number of steps in an optimization algorithm
M	Number of trainable parameters
n	Dimension of the input space $\boldsymbol{x} \in R^n$
N	Number of neurons, shallow network
N_i	Number of neurons, layer in a deep net
N	Bernstein N-width

$\mathcal{S}/\mathcal{S}_n$	Set of (shallow) neural networks
t	Time
w's	Weights (of a network)
W_m^n	Corresponding Sobolev space
$\mathbb{X}$	Function space (typically $\mathcal{C}^\infty$)
$\mathcal{X}$	Input space (finite version)
$\mathcal{Y}$	Output space (finite version)

Symbol	Use
α	Sometimes used as multi-index for derivatives
Δ	(Deep) network (as a function)
η	Neuron (individual, as a function)
η	Auxiliary function in mollifiers
ε	Small number
$\varphi, \varphi(t)$	Forgetting function
ν	Number of examples, size of $\boldsymbol{d}$
σ	Activation function
Σ	(Generally shallow) network (as a function)
τ	Time constant in exponential decay
Ω	Domain, $\Omega \subseteq \mathbb{R}^n$